

Social_Network_Analysis_Project

May 20, 2025

1 Dependencies

2 Imports

```
[1]: import networkx as nx
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from scipy import stats
import heapq
```

3 Environment Setup

4 Dataset Building

4.1 Downloading

```
[2]: def load_game_of_thrones_data():
    # Lista dei nomi dei file CSV
    file_names = ['book1.csv', 'book2.csv', 'book3.csv', 'book4.csv', 'book5.
    ↵csv']

    # Dizionario per memorizzare i dataframe
    dataframes = {}

    # Itera attraverso i file e carica i dati
    for file_name in file_names:
        url = f'https://raw.githubusercontent.com/RiccardoRomeo01/
    ↵SocialNetworkAnalysisProject/main/GameOfThronesDataset/{file_name}'
        dataframes[file_name] = pd.read_csv(url)

    return dataframes
```

```
[3]: # Utilizzo della funzione
dataframes = load_game_of_thrones_data()
```

```
# Accesso ai dataframe
book1_df = dataframes['book1.csv']
book2_df = dataframes['book2.csv']
book3_df = dataframes['book3.csv']
book4_df = dataframes['book4.csv']
book5_df = dataframes['book5.csv']
```

4.2 Exploring

[4] :

```
book1_df = book1_df.drop(labels=['book', 'Type'], axis=1)
book2_df = book2_df.drop(labels=['book', 'Type'], axis=1)
book3_df = book3_df.drop(labels=['book', 'Type'], axis=1)
book4_df = book4_df.drop(labels=['book', 'Type'], axis=1)
book5_df = book5_df.drop(labels=['book', 'Type'], axis=1)
```

[5] :

```
book1_df.head()
```

[5] :

	Source	Target	weight
0	Addam-Marbrand	Jaime-Lannister	3
1	Addam-Marbrand	Tywin-Lannister	6
2	Aegon-I-Targaryen	Daenerys-Targaryen	5
3	Aegon-I-Targaryen	Eddard-Stark	4
4	Aemon-Targaryen-(Maester-Aemon)	Alliser-Thorne	4

[6] :

```
all_book_df = pd.concat([book1_df, book2_df, book3_df, book4_df, book5_df])
all_book_df['Source'], all_book_df['Target'] = zip(*all_book_df.apply(lambda u
    ↪row: sorted([row['Source'], row['Target']]), axis=1))
```

5 Graph Creation

[7] :

```
G_1 = nx.from_pandas_edgelist(book1_df, 'Source', 'Target', edge_attr=True)
G_2 = nx.from_pandas_edgelist(book2_df, 'Source', 'Target', edge_attr=True)
G_3 = nx.from_pandas_edgelist(book3_df, 'Source', 'Target', edge_attr=True)
G_4 = nx.from_pandas_edgelist(book4_df, 'Source', 'Target', edge_attr=True)
G_5 = nx.from_pandas_edgelist(book5_df, 'Source', 'Target', edge_attr=True)
G_all = nx.from_pandas_edgelist(all_book_df, 'Source', 'Target', edge_attr=True)
```

6 Metrics

6.1 Nodes

6.1.1 Degree Centrality

```
[8]: def plot_degree_centrality(centrality, title, top_n=None, font_size=8, bar_width=1, spacing=1):
    """
    Plots the degree centrality of the given graph, showing only the top N elements if provided.

    Parameters:
    centrality (dict): The input graph centrality.
    title (str): The title for the plot.
    top_n (int or None): If None, all values are plotted. If an integer, only the top N values are plotted.
    font_size (int): The font size for the node labels in the plot.
    bar_width (float): The width of the bars in the histogram.
    spacing (float): The space between the bars.

    Returns:
    None: The function displays a plot of degree centrality.
    """
    # Prepare data for plotting
    nodes = list(centrality.keys())
    values = list(centrality.values())

    # Sort values and get top N if specified
    if top_n is not None:
        sorted_indices = np.argsort(values)[-top_n:] # Get indices of top N values
        filtered_nodes = [nodes[i] for i in sorted_indices]
        filtered_values = [values[i] for i in sorted_indices]
    else:
        filtered_nodes = nodes
        filtered_values = values

    # Create x-coordinates for bars with spacing
    x_positions = np.arange(len(filtered_nodes)) * (bar_width + spacing)

    # Create the plot
    plt.figure(figsize=(15, 5))
    bars = plt.bar(x_positions, filtered_values, color='skyblue', width=bar_width) # Adjust bar positions here
    plt.xlabel('Nodes')
    plt.ylabel('Degree Centrality')
```

```

plt.title(title, color="olivedrab")
plt.xticks(x_positions + bar_width / 2, filtered_nodes, rotation=90, u
˓→fontsize=font_size) # Center labels
plt.grid(axis='y')
plt.xlim(-0.5, len(filtered_nodes) * (bar_width + spacing) - spacing) # u
˓→Limiti dell'asse x

# Annotate bars with their values
for bar in bars:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, round(yval, 2), u
˓→ha='center', va='bottom', fontsize=font_size)

# Show the plot
plt.tight_layout()
plt.show()

return dict(zip(filtered_nodes, filtered_values))

```

```

[ ]: def common_keys(list_of_dict):
    """
    Restituisce una lista delle chiavi comuni tra tutti i dizionari in input.

    Parameters:
    list_of_dict (list): Una lista di dizionari.

    Returns:
    list: Una lista delle chiavi comuni.
    """
    if not list_of_dict:
        return []

    # Inizializza le chiavi comuni con le chiavi del primo dizionario
    chiavi_comuni = set(list_of_dict[0].keys())

    # Itera attraverso gli altri dizionari e aggiorna le chiavi comuni
    for dizionario in list_of_dict[1:]:
        chiavi_comuni.intersection_update(dizionario.keys())

    return list(chiavi_comuni)

```

```

[ ]: def show_evolution(dizionari_libri, personaggi, xlabel="Books", ylabel="", u
˓→title=""):
    """
    Shows the evolution of character grades across the 5 books of Game of u
˓→Thrones.

```

Parameters:

dizionari_libri (*list*): A list of 5 dictionaries, each containing the ↪grades of characters for one book.

Each dictionary should have the structure:
 $\{'character': grade\}$

personaggi (*list*): A list of characters to display.

title (*str*): The title of the graph.

Returns:

None: The function displays the graph.

"""

```

# Create a list of grades for each character
gradi = {personaggio: [] for personaggio in personaggi}

# Extract grades from the dictionaries
for dizionario in dizionari_libri:
    for personaggio in personaggi:
        grado = dizionario.get(personaggio)
        gradi[personaggio].append(grado if grado is not None else 0) # Use ↪0 if the character is not present

# Create the graph
plt.figure(figsize=(12, 6))

for personaggio in personaggi:
    plt.plot(gradi[personaggio], marker='o', label=personaggio)

# Add labels and title
plt.title(title, color="olivedrab")
plt.xlabel(xlabel)
plt.ylabel(ylabel)
plt.xticks(ticks=range(5), labels=[f'Book {i+1}' for i in range(5)])
plt.legend()
plt.grid(axis='y')

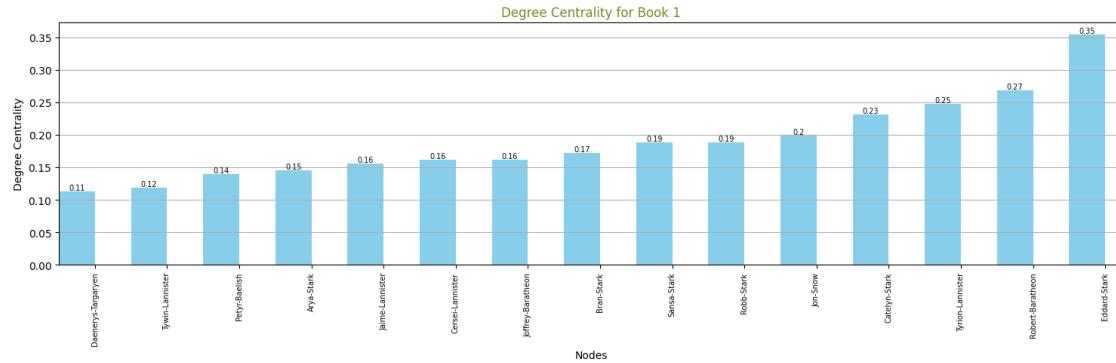
# Show the graph
plt.tight_layout()
plt.show()
```

[]:

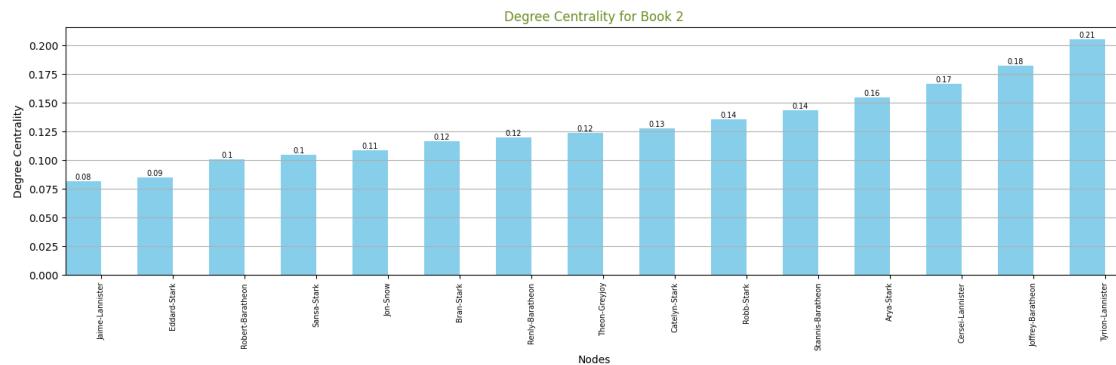
```

degree_1 = nx.degree_centrality(G_1)
degree_2 = nx.degree_centrality(G_2)
degree_3 = nx.degree_centrality(G_3)
degree_4 = nx.degree_centrality(G_4)
degree_5 = nx.degree_centrality(G_5)
degree_all = nx.degree_centrality(G_all)
```

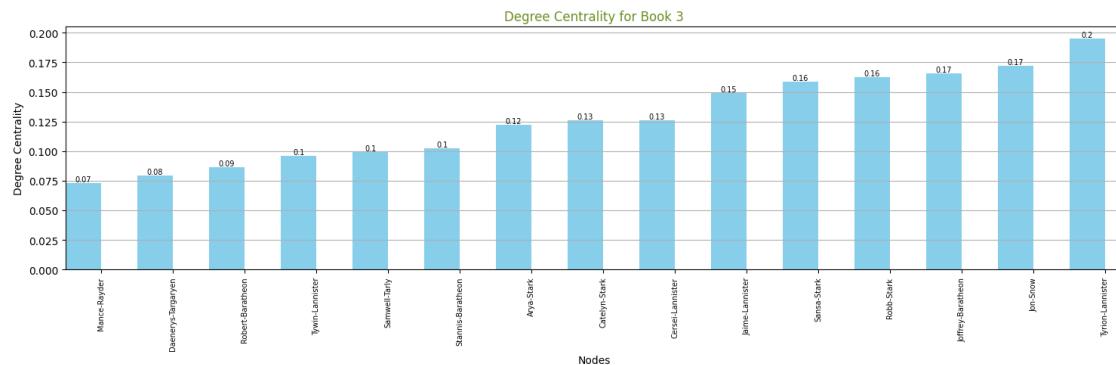
```
[ ]: top_degree_nodes_1 = plot_degree_centrality(degree_1, 'Degree Centrality for Book 1', font_size=7, top_n=15)
```



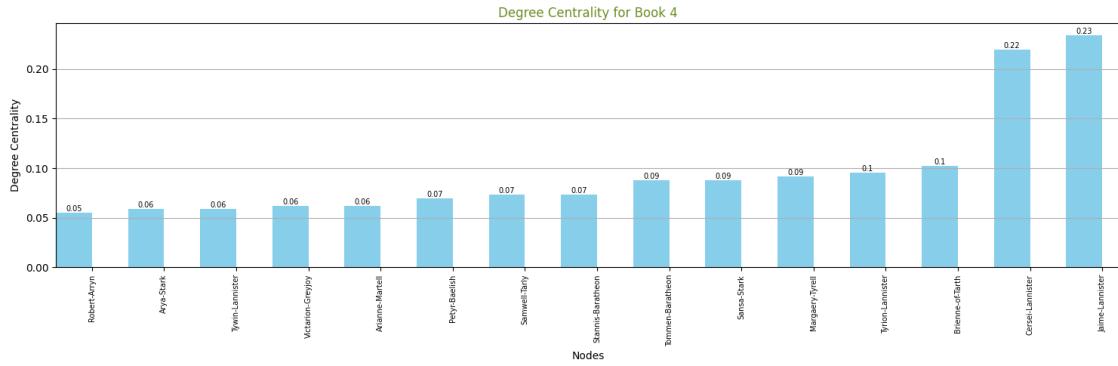
```
[ ]: top_degree_nodes_2 = plot_degree_centrality(degree_2, 'Degree Centrality for Book 2', font_size=7, top_n=15)
```



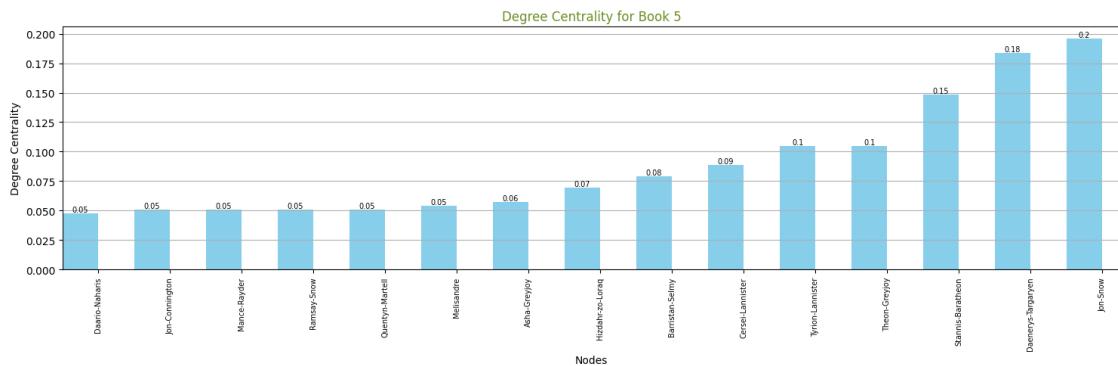
```
[ ]: top_degree_nodes_3 = plot_degree_centrality(degree_3, 'Degree Centrality for Book 3', font_size=7, top_n=15)
```



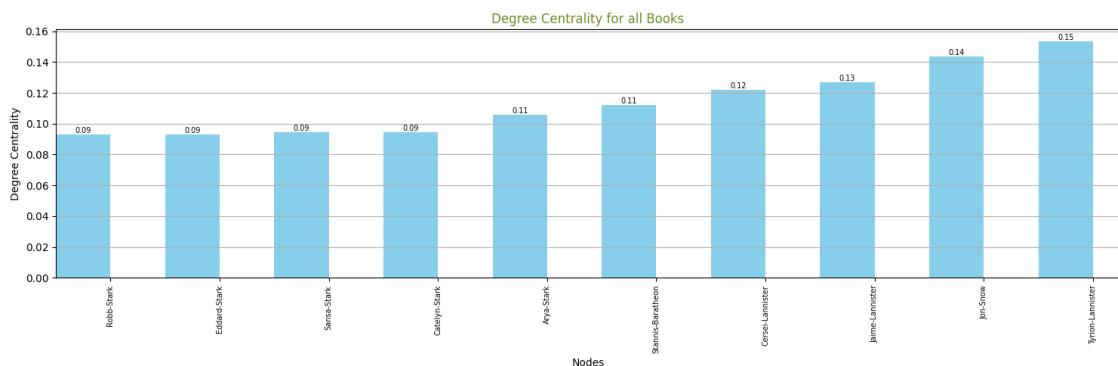
```
[ ]: top_degree_nodes_4 = plot_degree_centrality(degree_4, 'Degree Centrality for Book 4', font_size=7, top_n=15)
```



```
[ ]: top_degree_nodes_5 = plot_degree_centrality(degree_5, 'Degree Centrality for Book 5', font_size=7, top_n=15)
```



```
[ ]: top_degree_nodes_all = plot_degree_centrality(degree_all, 'Degree Centrality for all Books', font_size=7, top_n=10)
```

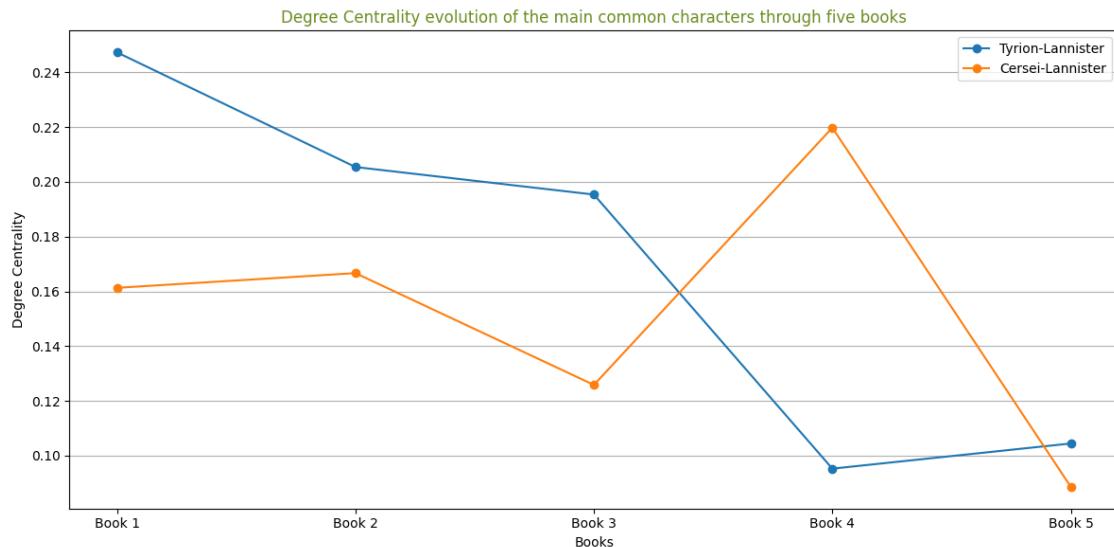


```
[ ]: top_common_nodes_DC = common_keys([top_degree_nodes_1, top_degree_nodes_2,
                                         ↪top_degree_nodes_3, top_degree_nodes_4, top_degree_nodes_5])

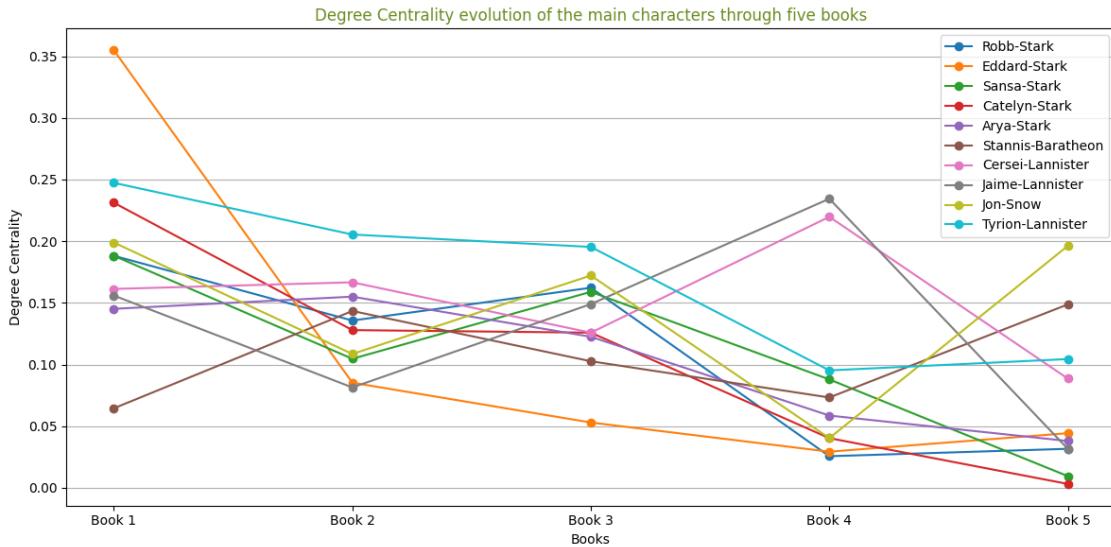
[ ]: top_common_nodes_DC

[ ]: ['Tyrion-Lannister', 'Cersei-Lannister']

[ ]: show_evolution([top_degree_nodes_1, top_degree_nodes_2, top_degree_nodes_3,
                     ↪top_degree_nodes_4, top_degree_nodes_5],
                  top_common_nodes_DC,
                  title="Degree Centrality evolution of the main common
                     ↪characters through five books",
                  xlabel="Books",
                  ylabel="Degree Centrality")
```



```
[ ]: show_evolution([degree_1, degree_2, degree_3, degree_4, degree_5],
                  top_degree_nodes_all,
                  title="Degree Centrality evolution of the main
                     ↪characters through five books",
                  xlabel="Books",
                  ylabel="Degree Centrality")
```



6.1.2 Eigenvector Centrality

```
[ ]: def plot_eig_centrality(centrality, title, top_n=None, font_size=8, bar_width=1, spacing=1):
    """
    Plots the Eigenvector centrality of the given graph, showing only the top N elements if provided.

    Parameters:
    centrality (dict): The input graph centrality.
    title (str): The title for the plot.
    top_n (int or None): If None, all values are plotted. If an integer, only the top N values are plotted.
    font_size (int): The font size for the node labels in the plot.
    bar_width (float): The width of the bars in the histogram.
    spacing (float): The space between the bars.

    Returns:
    None: The function displays a plot of eigenvector centrality.

    # Prepare data for plotting
    nodes = list(centrality.keys())
    values = list(centrality.values())

    # Sort values and get top N if specified
    if top_n is not None:
```

```

    sorted_indices = np.argsort(values)[-top_n:] # Get indices of top N
    ↪values

    filtered_nodes = [nodes[i] for i in sorted_indices]
    filtered_values = [values[i] for i in sorted_indices]

else:
    filtered_nodes = nodes
    filtered_values = values

# Create x-coordinates for bars with spacing
x_positions = np.arange(len(filtered_nodes)) * (bar_width + spacing)

# Create the plot
plt.figure(figsize=(15, 5))
bars = plt.bar(x_positions, filtered_values, color='skyblue', ↪
    ↪width=bar_width) # Adjust bar positions here
plt.xlabel('Nodes')
plt.ylabel('Eigenvector Centrality')
plt.title(title, color="olivedrab")
plt.xticks(x_positions + bar_width / 2, filtered_nodes, rotation=90, ↪
    ↪fontsize=font_size) # Center labels
plt.grid(axis='y')
plt.xlim(-0.5, len(filtered_nodes) * (bar_width + spacing) - spacing) # ↪
    ↪Limiti dell'asse x

# Annotate bars with their values
for bar in bars:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, round(yval, 2), ↪
        ↪ha='center', va='bottom', fontsize=font_size)

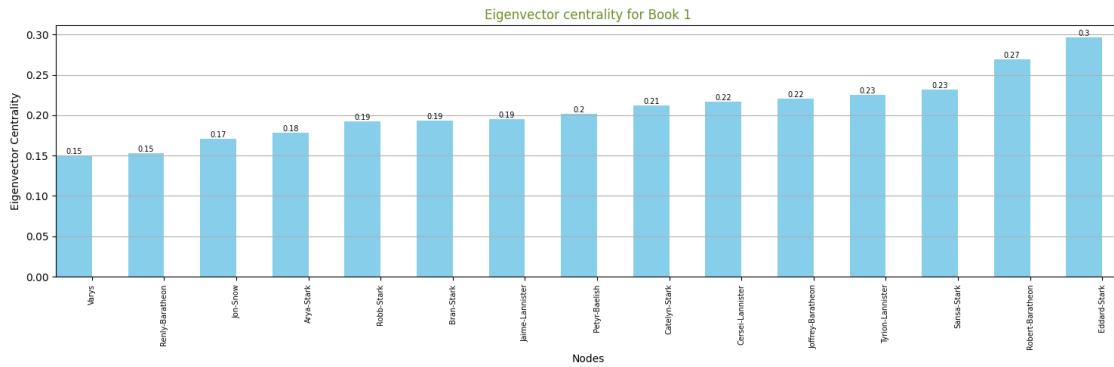
# Show the plot
plt.tight_layout()
plt.show()

return dict(zip(filtered_nodes, filtered_values))

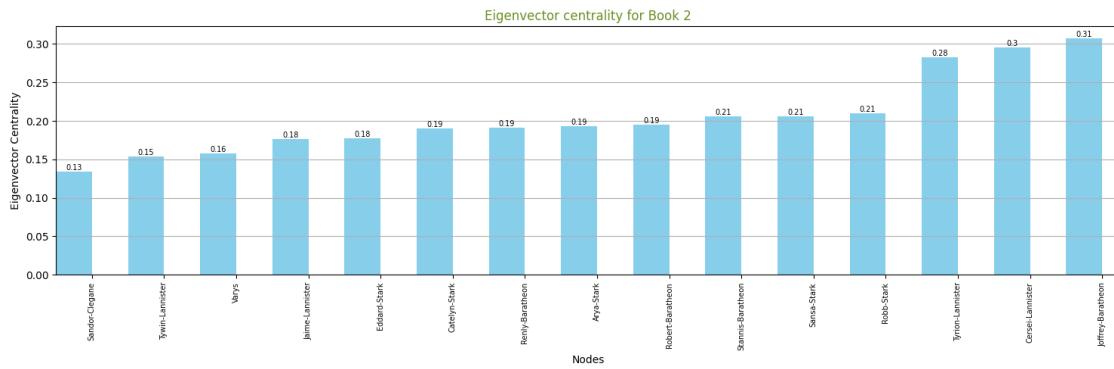
```

```
[ ]: eig_1 = nx.eigenvector_centrality(G_1, weight='Weight')
eig_2 = nx.eigenvector_centrality(G_2, weight='Weight')
eig_3 = nx.eigenvector_centrality(G_3, weight='Weight')
eig_4 = nx.eigenvector_centrality(G_4, weight='Weight')
eig_5 = nx.eigenvector_centrality(G_5, weight='Weight', max_iter=200)
eig_all = nx.eigenvector_centrality(G_all, weight='Weight')
```

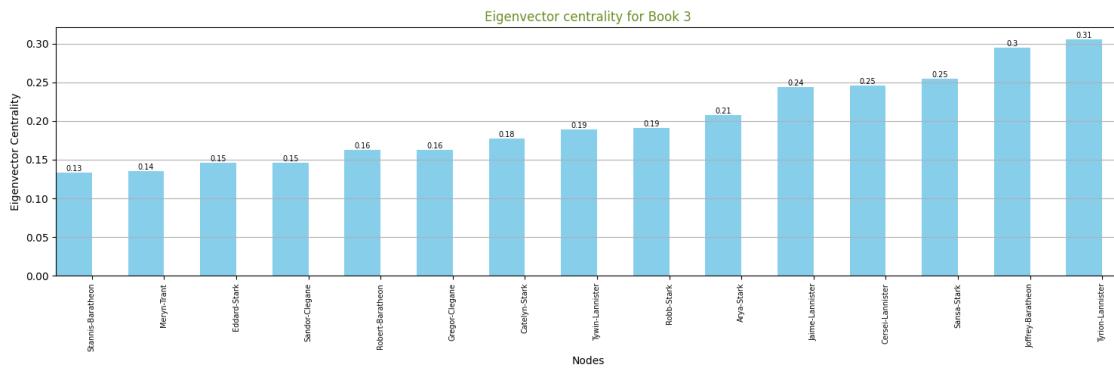
```
[ ]: top_eig_nodes_1 = plot_eig_centrality(eig_1, 'Eigenvector centrality for Book ↪
    ↪1', font_size=7, top_n=15)
```



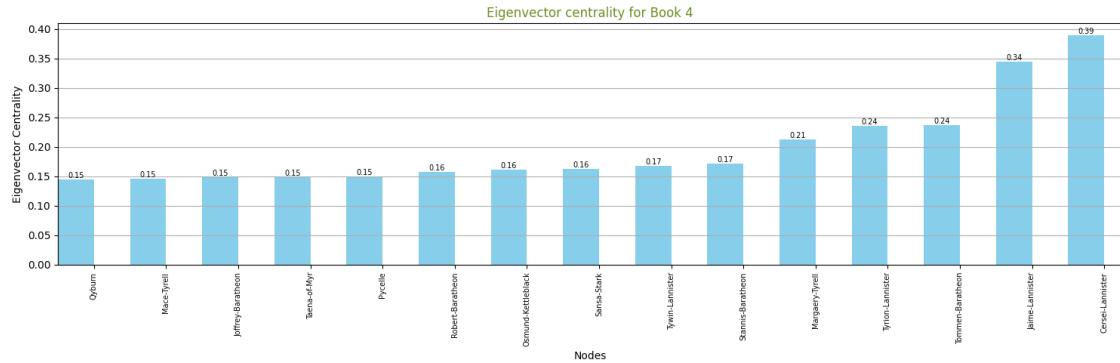
```
[ ]: top_eig_nodes_2 = plot_eig_centrality(eig_2, 'Eigenvector centrality for Book 2', font_size=7, top_n=15)
```



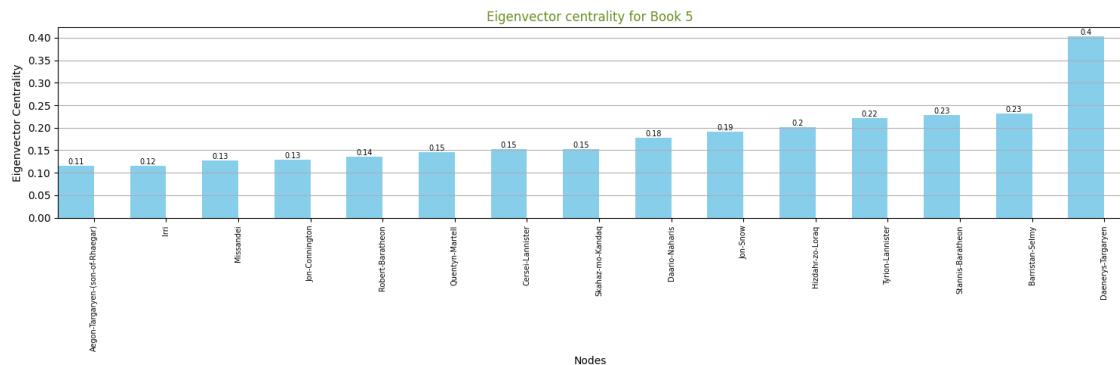
```
[ ]: top_eig_nodes_3 = plot_eig_centrality(eig_3, 'Eigenvector centrality for Book 3', font_size=7, top_n=15)
```



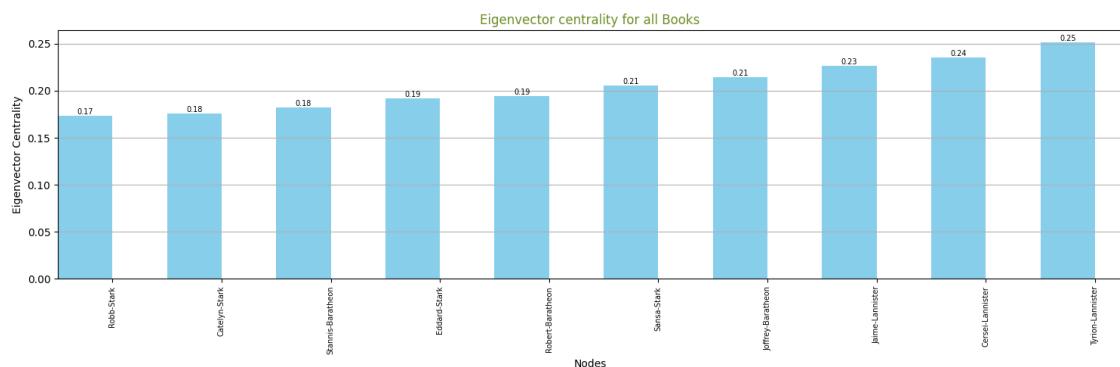
```
[ ]: top_eig_nodes_4 = plot_eig_centrality(eig_4, 'Eigenvector centrality for Book 4', font_size=7, top_n=15)
```



```
[ ]: top_eig_nodes_5 = plot_eig_centrality(eig_5, 'Eigenvector centrality for Book 5', font_size=7, top_n=15)
```



```
[ ]: top_eig_nodes_all = plot_eig_centrality(eig_all, 'Eigenvector centrality for all Books', font_size=7, top_n=10)
```

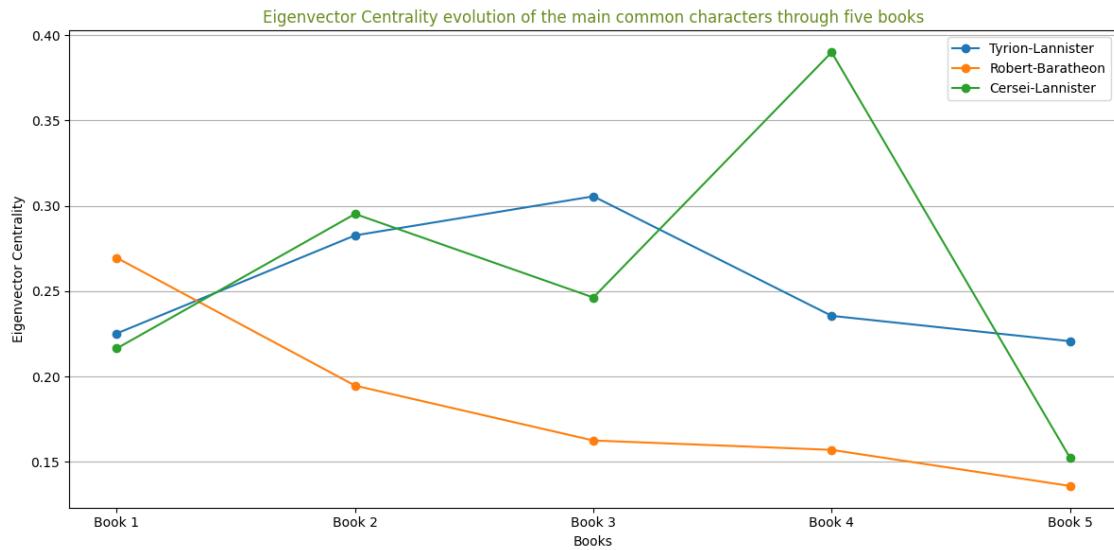


```
[ ]: top_common_nodes_EVC = common_keys([top_eig_nodes_1, top_eig_nodes_2, top_eig_nodes_3, top_eig_nodes_4, top_eig_nodes_5])

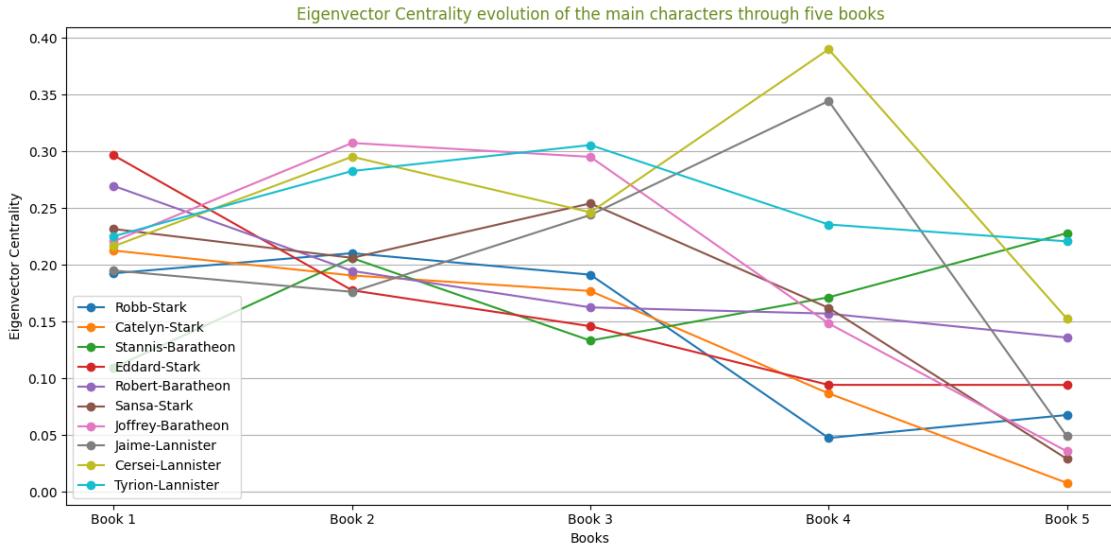
[ ]: top_common_nodes_EVC

[ ]: ['Tyrion-Lannister', 'Robert-Baratheon', 'Cersei-Lannister']

[ ]: show_evolution([top_eig_nodes_1, top_eig_nodes_2, top_eig_nodes_3, top_eig_nodes_4, top_eig_nodes_5],
                  top_common_nodes_EVC,
                  title="Eigenvector Centrality evolution of the main common characters through five books",
                  xlabel="Books",
                  ylabel="Eigenvector Centrality")
```



```
[ ]: show_evolution([eig_1, eig_2, eig_3, eig_4, eig_5],
                  top_eig_nodes_all,
                  title="Eigenvector Centrality evolution of the main characters through five books",
                  xlabel="Books",
                  ylabel="Eigenvector Centrality")
```



6.1.3 Katz Centrality (Page Rank)

```
[ ]: def plot_katz_centrality(centrality, title, top_n=None, font_size=8, bar_width=1, spacing=1):
    """
    Plots the Katz centrality of the given graph, showing only the top N elements if provided.

    Parameters:
    centrality (dict): The input graph centrality.
    title (str): The title for the plot.
    top_n (int or None): If None, all values are plotted. If an integer, only the top N values are plotted.
    font_size (int): The font size for the node labels in the plot.
    bar_width (float): The width of the bars in the histogram.
    spacing (float): The space between the bars.

    Returns:
    None: The function displays a plot of katz centrality.

    # Prepare data for plotting
    nodes = list(centrality.keys())
    values = list(centrality.values())

    # Sort values and get top N if specified
    if top_n is not None:
```

```

        sorted_indices = np.argsort(values)[-top_n:] # Get indices of top N
        ↵values

        filtered_nodes = [nodes[i] for i in sorted_indices]
        filtered_values = [values[i] for i in sorted_indices]

    else:
        filtered_nodes = nodes
        filtered_values = values

    # Create x-coordinates for bars with spacing
    x_positions = np.arange(len(filtered_nodes)) * (bar_width + spacing)

    # Create the plot
    plt.figure(figsize=(15, 5))
    bars = plt.bar(x_positions, filtered_values, color='skyblue', ↵
    ↵width=bar_width) # Adjust bar positions here
    plt.xlabel('Nodes')
    plt.ylabel('Katz Centrality')
    plt.title(title, color="olivedrab")
    plt.xticks(x_positions + bar_width / 2, filtered_nodes, rotation=90, ↵
    ↵fontsize=font_size) # Center labels
    plt.grid(axis='y')
    plt.xlim(-0.5, len(filtered_nodes) * (bar_width + spacing) - spacing) # ↵
    ↵Limiti dell'asse x

    # Annotate bars with their values
    for bar in bars:
        yval = bar.get_height()
        plt.text(bar.get_x() + bar.get_width()/2, yval, round(yval, 3), ↵
        ↵ha='center', va='bottom', fontsize=font_size)

    # Show the plot
    plt.tight_layout()
    plt.show()

    return dict(zip(filtered_nodes, filtered_values))

```

```

[ ]: max_eig = max(nx.adjacency_spectrum(G_1, weight='weight')).real
katz_1 = nx.katz_centrality_numpy(G_1, alpha = 1/(max_eig+1), weight='Weight')

max_eig = max(nx.adjacency_spectrum(G_2, weight='weight')).real
katz_2 = nx.katz_centrality_numpy(G_2, alpha = 1/(max_eig+1), weight='Weight')

max_eig = max(nx.adjacency_spectrum(G_3, weight='weight')).real
katz_3 = nx.katz_centrality_numpy(G_3, alpha = 1/(max_eig+1), weight='Weight')

max_eig = max(nx.adjacency_spectrum(G_4, weight='weight')).real
katz_4 = nx.katz_centrality_numpy(G_4, alpha = 1/(max_eig+1), weight='Weight')

```

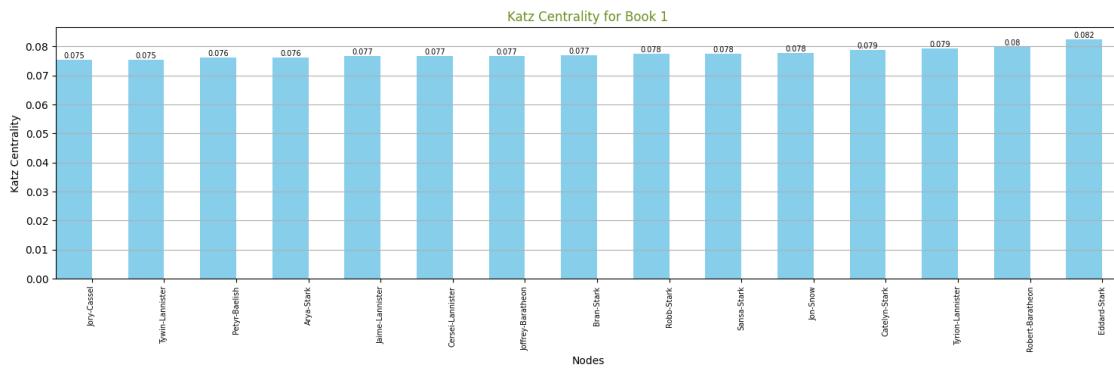
```

max_eig = max(nx.adjacency_spectrum(G_5, weight='weight')).real
katz_5 = nx.katz_centrality_numpy(G_5, alpha = 1/(max_eig+1), weight='Weight')

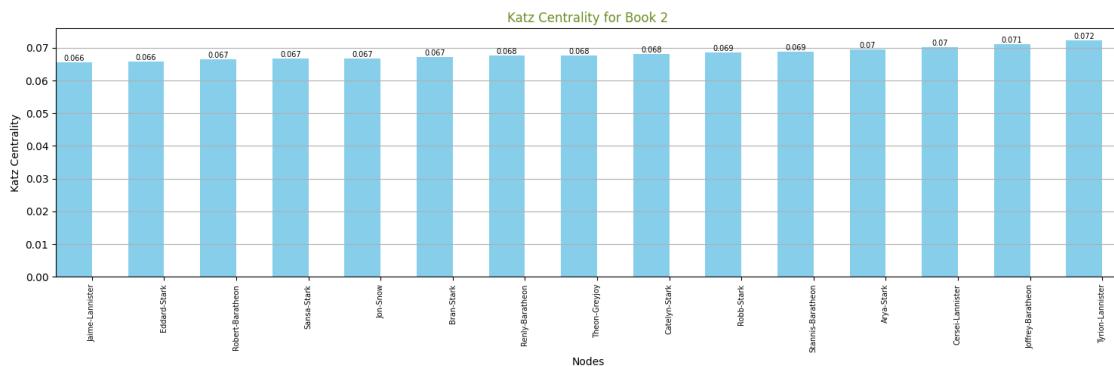
max_eig = max(nx.adjacency_spectrum(G_all, weight='weight')).real
katz_all = nx.katz_centrality_numpy(G_all, alpha = 1/(max_eig+1), weight='Weight')

```

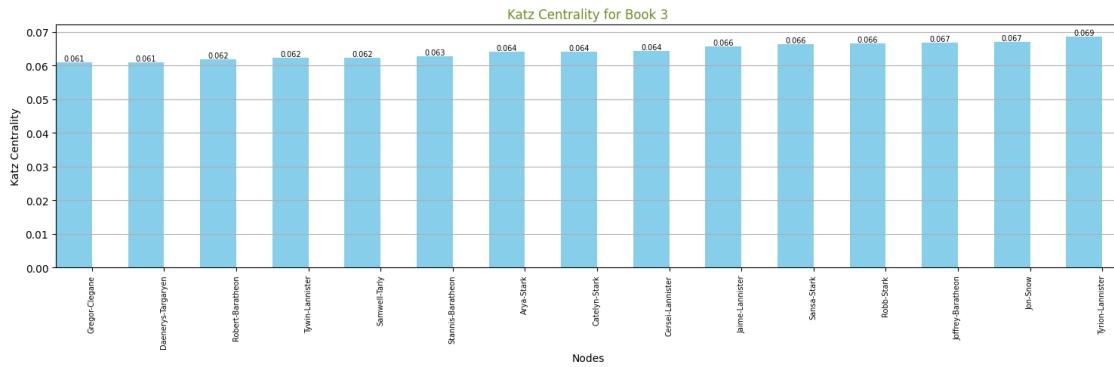
[]: `top_katz_nodes_1 = plot_katz_centrality(katz_1, 'Katz Centrality for Book 1', font_size=7, top_n=15)`



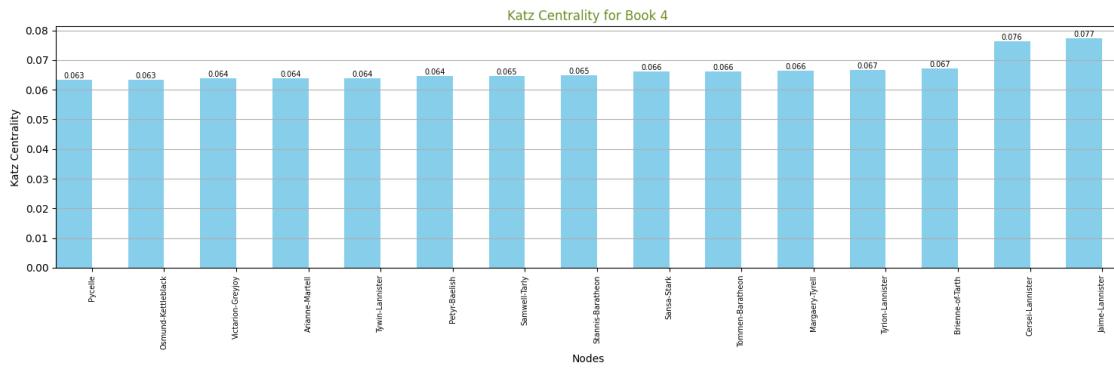
[]: `top_katz_nodes_2 = plot_katz_centrality(katz_2, 'Katz Centrality for Book 2', font_size=7, top_n=15)`



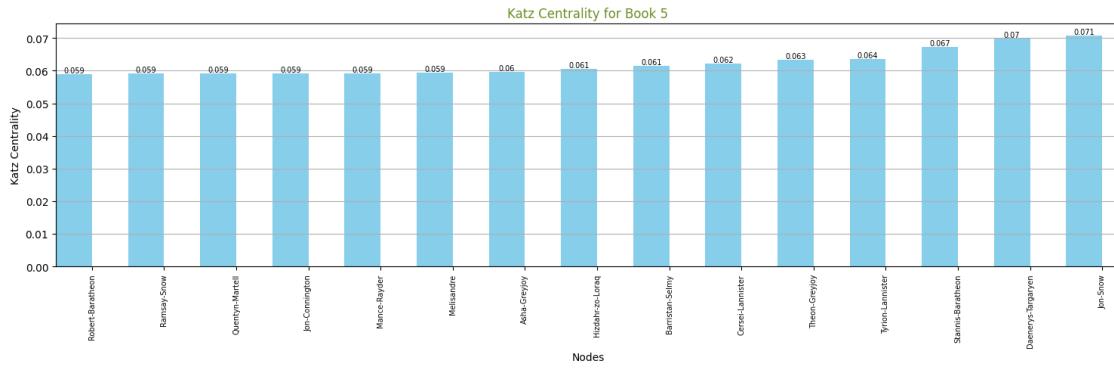
[]: `top_katz_nodes_3 = plot_katz_centrality(katz_3, 'Katz Centrality for Book 3', font_size=7, top_n=15)`



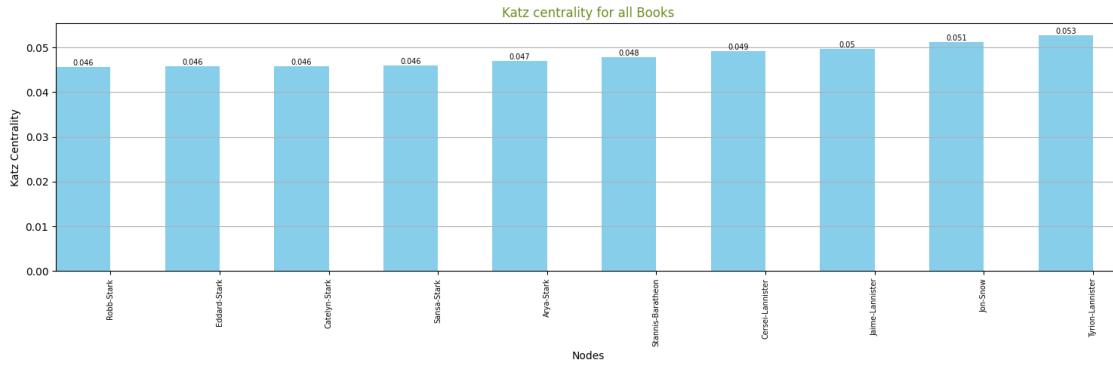
```
[ ]: top_katz_nodes_4 = plot_katz_centrality(katz_4, 'Katz Centrality for Book 4',  
    font_size=7, top_n=15)
```



```
[ ]: top_katz_nodes_5 = plot_katz_centrality(katz_5, 'Katz Centrality for Book 5',  
    font_size=7, top_n=15)
```



```
[ ]: top_katz_nodes_all = plot_katz_centrality(katz_all, 'Katz centrality for all Books', font_size=7, top_n=10)
```

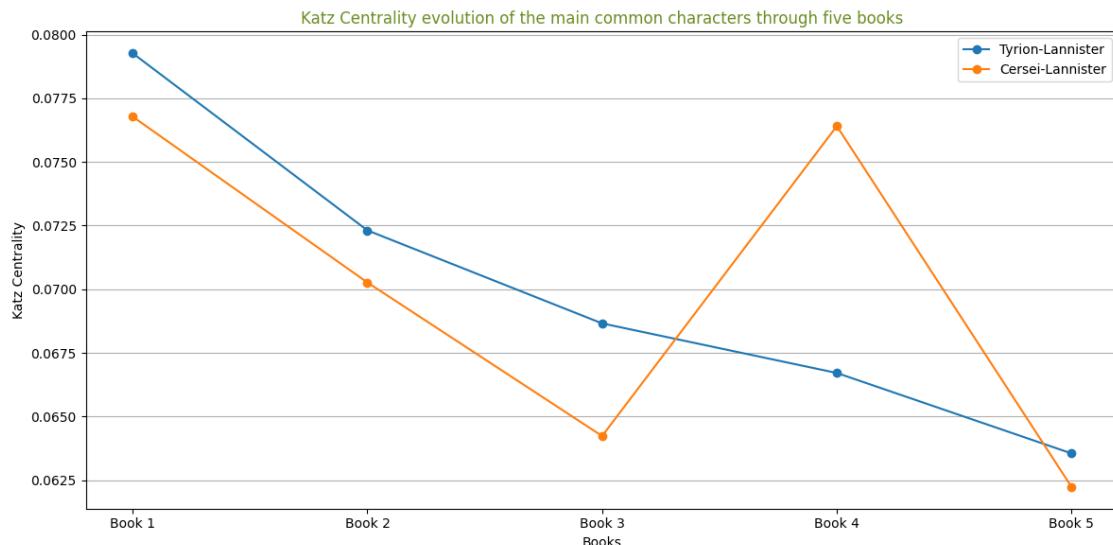


```
[ ]: top_common_nodes_KC = common_keys([top_katz_nodes_1, top_katz_nodes_2, top_katz_nodes_3, top_katz_nodes_4, top_katz_nodes_5])
```

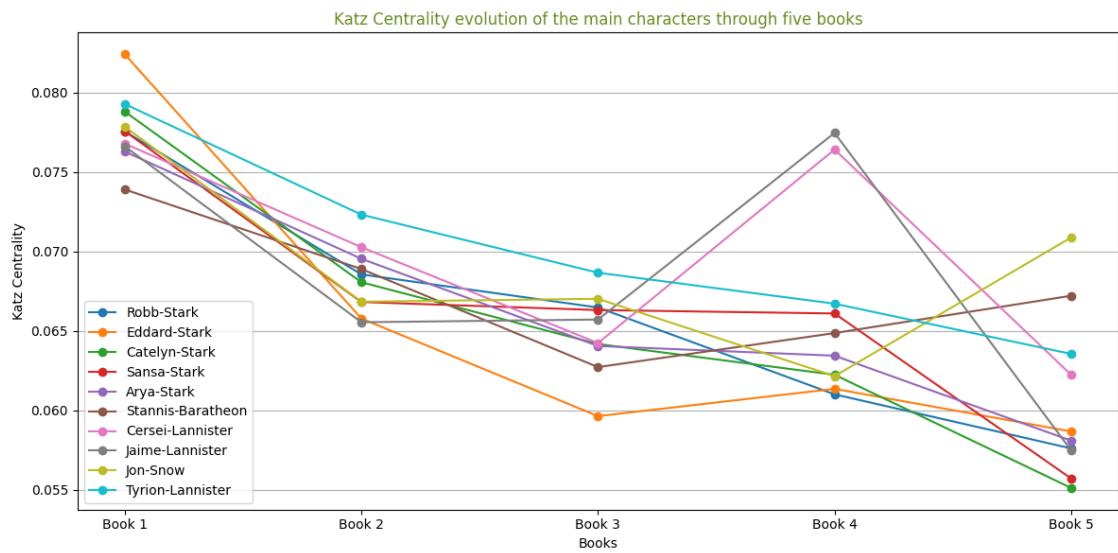
```
[ ]: top_common_nodes_KC
```

```
[ ]: ['Tyrion-Lannister', 'Cersei-Lannister']
```

```
[ ]: show_evolution([top_katz_nodes_1, top_katz_nodes_2, top_katz_nodes_3, top_katz_nodes_4, top_katz_nodes_5], top_common_nodes_KC, title="Katz Centrality evolution of the main common characters through five books", xlabel="Books", ylabel="Katz Centrality")
```



```
[ ]: show_evolution([katz_1, katz_2, katz_3, katz_4, katz_5],
                  top_katz_nodes_all,
                  title="Katz Centrality evolution of the main characters through five books",
                  xlabel="Books",
                  ylabel="Katz Centrality")
```



6.1.4 Closeness Centrality

```
[ ]: def plot_closeness_centrality(centrality, title, top_n=None, font_size=8,
                                bar_width=1, spacing=1):
    """
    Plots the Closeness centrality of the given graph, showing only the top N elements if provided.
    """
```

Parameters:

centrality (dict): The input graph centrality.

title (str): The title for the plot.

top_n (int or None): If None, all values are plotted. If an integer, only the top N values are plotted.

font_size (int): The font size for the node labels in the plot.

bar_width (float): The width of the bars in the histogram.

spacing (float): The space between the bars.

Returns:

None: The function displays a plot of closeness centrality.

```

"""
# Prepare data for plotting
nodes = list(centrality.keys())
values = list(centrality.values())

# Sort values and get top N if specified
if top_n is not None:
    sorted_indices = np.argsort(values)[-top_n:] # Get indices of top N
    ↪values
    filtered_nodes = [nodes[i] for i in sorted_indices]
    filtered_values = [values[i] for i in sorted_indices]
else:
    filtered_nodes = nodes
    filtered_values = values

# Create x-coordinates for bars with spacing
x_positions = np.arange(len(filtered_nodes)) * (bar_width + spacing)

# Create the plot
plt.figure(figsize=(15, 5))
bars = plt.bar(x_positions, filtered_values, color='skyblue', ↪
    ↪width=bar_width) # Adjust bar positions here
plt.xlabel('Nodes')
plt.ylabel('Closeness Centrality')
plt.title(title, color="olivedrab")
plt.xticks(x_positions + bar_width / 2, filtered_nodes, rotation=90, ↪
    ↪fontsize=font_size) # Center labels
plt.grid(axis='y')
plt.xlim(-0.5, len(filtered_nodes) * (bar_width + spacing) - spacing) # ↪
    ↪Limiti dell'asse x

# Annotate bars with their values
for bar in bars:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, round(yval, 3), ↪
        ↪ha='center', va='bottom', fontsize=font_size)

# Show the plot
plt.tight_layout()
plt.show()

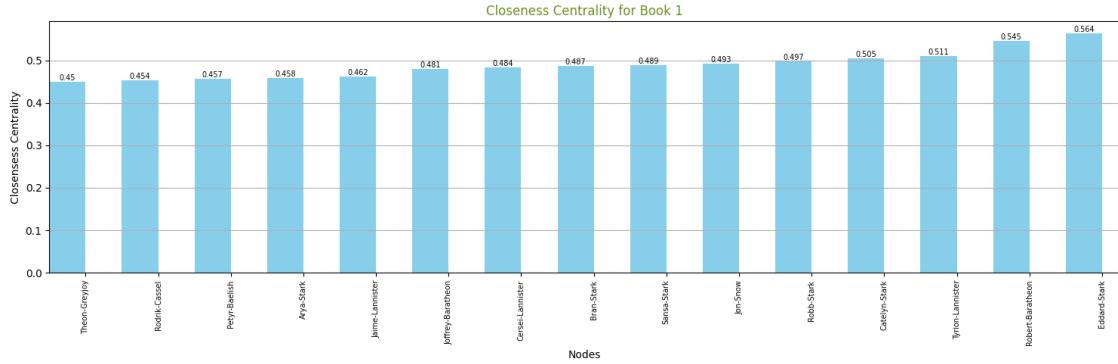
return dict(zip(filtered_nodes, filtered_values))

```

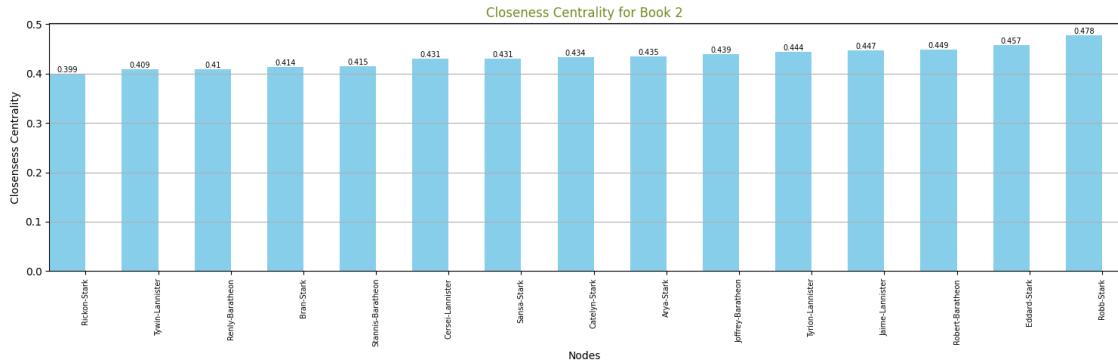
```
[ ]: close_1 = nx.closeness_centrality(G_1, u=None, distance=1)
close_2 = nx.closeness_centrality(G_2, u=None, distance=1)
close_3 = nx.closeness_centrality(G_3, u=None, distance=1)
```

```
close_4 = nx.closeness_centrality(G_4, u=None, distance=1)
close_5 = nx.closeness_centrality(G_5, u=None, distance=1)
close_all = nx.closeness_centrality(G_all, u=None, distance=1)
```

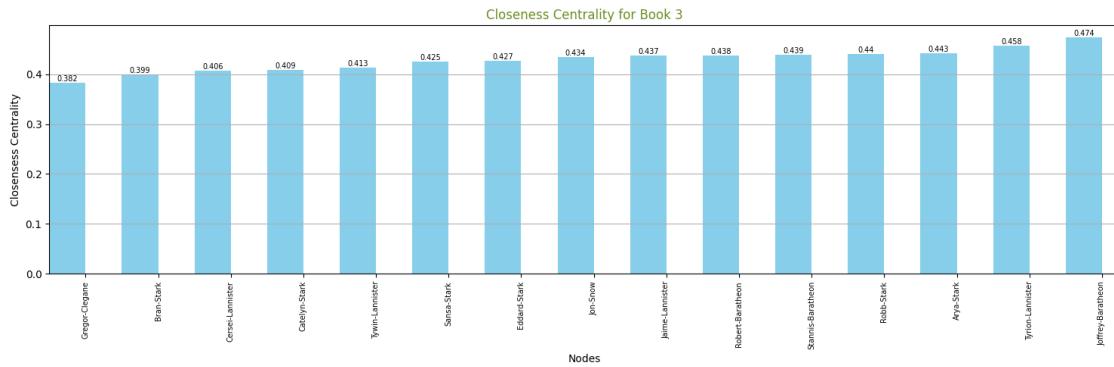
[]: top_close_nodes_1 = plot_closeness_centrality(close_1, 'Closeness Centrality for Book 1', font_size=7, top_n=15)



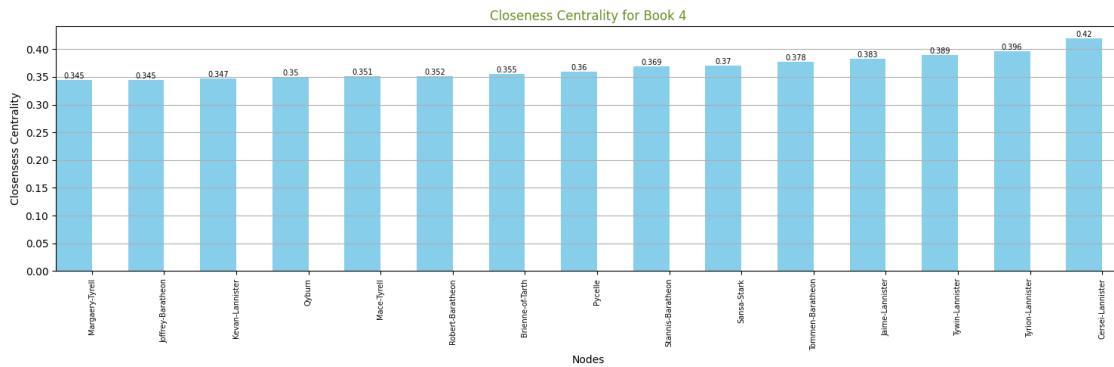
[]: top_close_nodes_2 = plot_closeness_centrality(close_2, 'Closeness Centrality for Book 2', font_size=7, top_n=15)



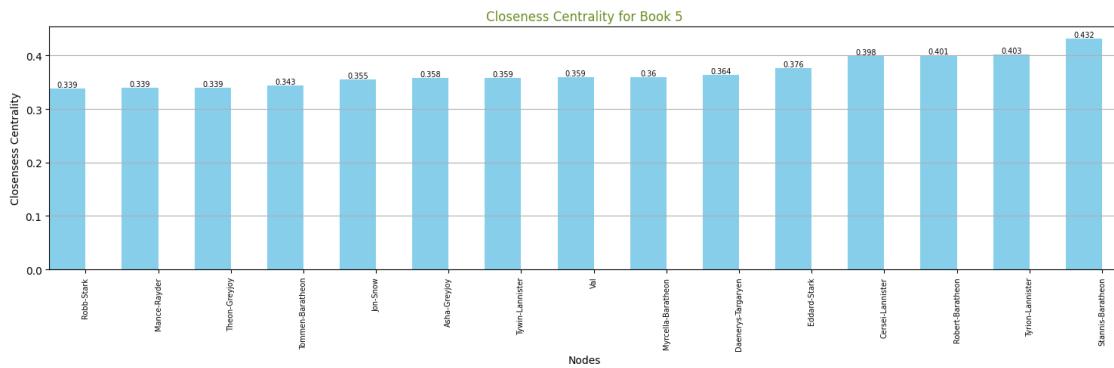
[]: top_close_nodes_3 = plot_closeness_centrality(close_3, 'Closeness Centrality for Book 3', font_size=7, top_n=15)



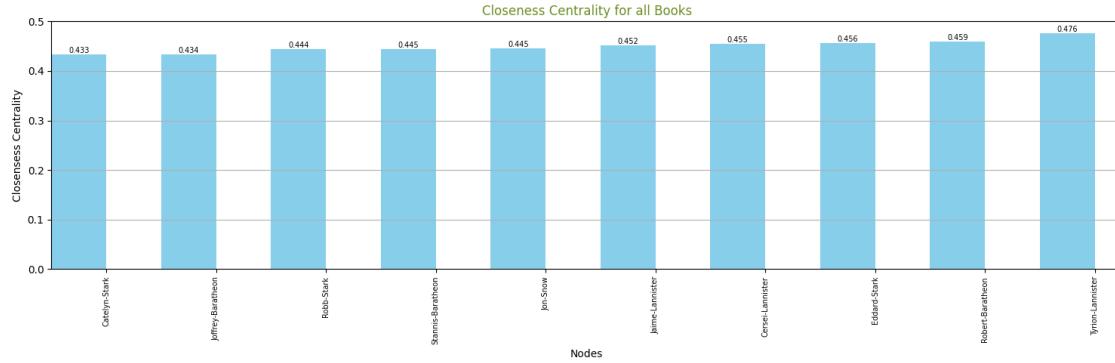
```
[ ]: top_close_nodes_4 = plot_closeness_centrality(close_4, 'Closeness Centrality for Book 4', font_size=7, top_n=15)
```



```
[ ]: top_close_nodes_5 = plot_closeness_centrality(close_5, 'Closeness Centrality for Book 5', font_size=7, top_n=15)
```



```
[ ]: top_close_nodes_all = plot_closeness_centrality(close_all, 'Closeness Centrality for all Books', font_size=7, top_n=10)
```

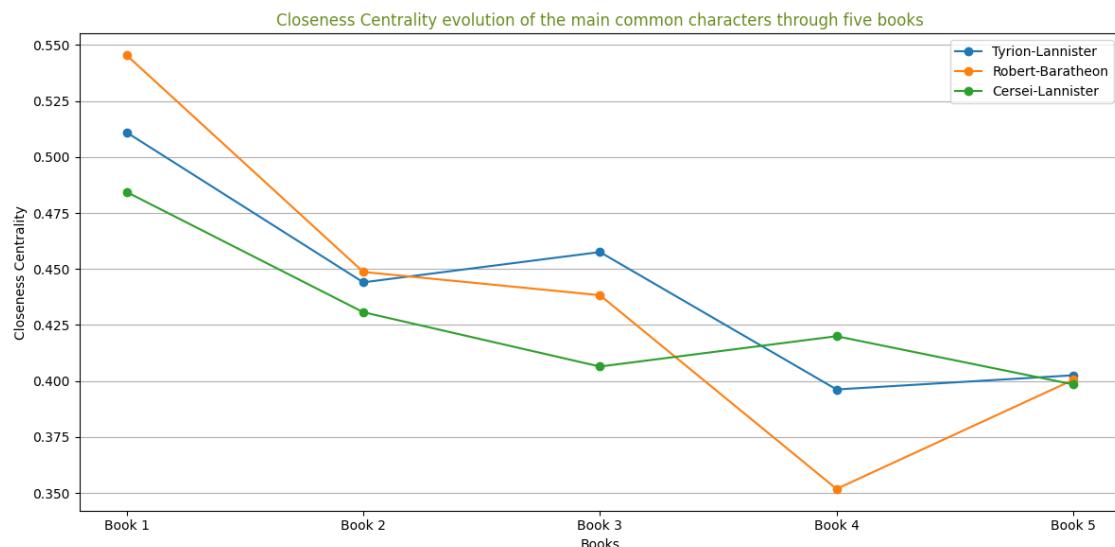


```
[ ]: top_common_nodes_CC = common_keys([top_close_nodes_1, top_close_nodes_2, top_close_nodes_3, top_close_nodes_4, top_close_nodes_5])
```

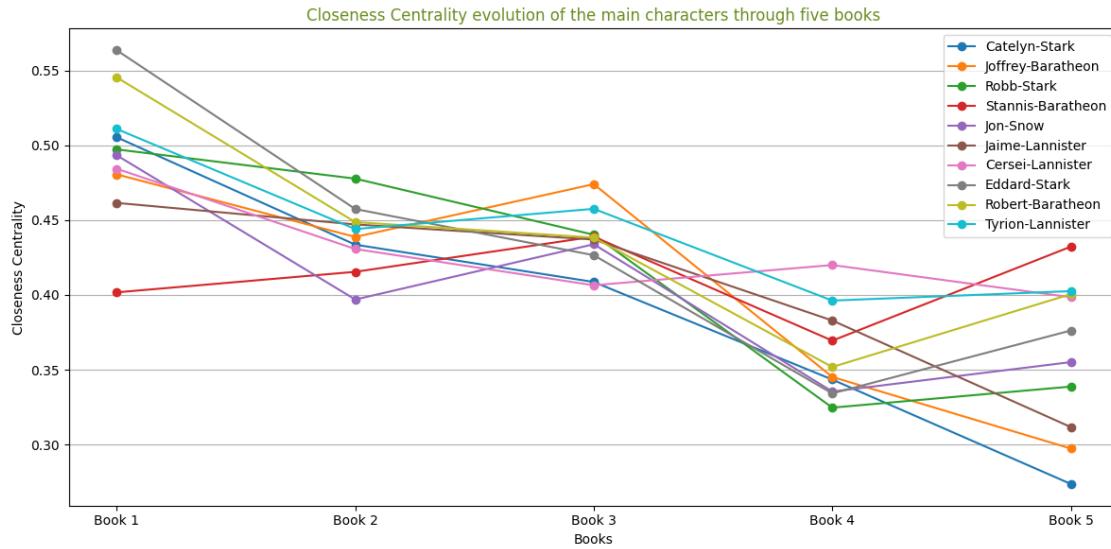
```
[ ]: top_common_nodes_CC
```

```
[ ]: ['Tyrion-Lannister', 'Robert-Baratheon', 'Cersei-Lannister']
```

```
[ ]: show_evolution([top_close_nodes_1, top_close_nodes_2, top_close_nodes_3, top_close_nodes_4, top_close_nodes_5], top_common_nodes_CC, title="Closeness Centrality evolution of the main common characters through five books", xlabel="Books", ylabel="Closeness Centrality")
```



```
[ ]: show_evolution([close_1, close_2, close_3, close_4, close_5],
                  top_close_nodes_all,
                  title="Closeness Centrality evolution of the main characters through five books",
                  xlabel="Books",
                  ylabel="Closeness Centrality")
```



6.1.5 Betweenness Centrality

```
[ ]: def plot_betweenness_centrality(centrality, title, top_n=None, font_size=8, bar_width=1, spacing=1):
    """
    Plots the Betweenness centrality of the given graph, showing only the top N elements if provided.

    Parameters:
    centrality (dict): The input graph centrality.
    title (str): The title for the plot.
    top_n (int or None): If None, all values are plotted. If an integer, only the top N values are plotted.
    font_size (int): The font size for the node labels in the plot.
    bar_width (float): The width of the bars in the histogram.
    spacing (float): The space between the bars.

    Returns:
    None: The function displays a plot of Betweenness centrality.
```

```

"""
# Prepare data for plotting
nodes = list(centrality.keys())
values = list(centrality.values())

# Sort values and get top N if specified
if top_n is not None:
    sorted_indices = np.argsort(values)[-top_n:] # Get indices of top N
    ↪values
    filtered_nodes = [nodes[i] for i in sorted_indices]
    filtered_values = [values[i] for i in sorted_indices]
else:
    filtered_nodes = nodes
    filtered_values = values

# Create x-coordinates for bars with spacing
x_positions = np.arange(len(filtered_nodes)) * (bar_width + spacing)

# Create the plot
plt.figure(figsize=(15, 5))
bars = plt.bar(x_positions, filtered_values, color='skyblue', ↪
    ↪width=bar_width) # Adjust bar positions here
plt.xlabel('Nodes')
plt.ylabel('Closeness Centrality')
plt.title(title, color="olivedrab")
plt.xticks(x_positions + bar_width / 2, filtered_nodes, rotation=90, ↪
    ↪fontsize=font_size) # Center labels
plt.grid(axis='y')
plt.xlim(-0.5, len(filtered_nodes) * (bar_width + spacing) - spacing) # ↪
    ↪Limiti dell'asse x

# Annotate bars with their values
for bar in bars:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, round(yval, 3), ↪
        ↪ha='center', va='bottom', fontsize=font_size)

# Show the plot
plt.tight_layout()
plt.show()

return dict(zip(filtered_nodes, filtered_values))

```

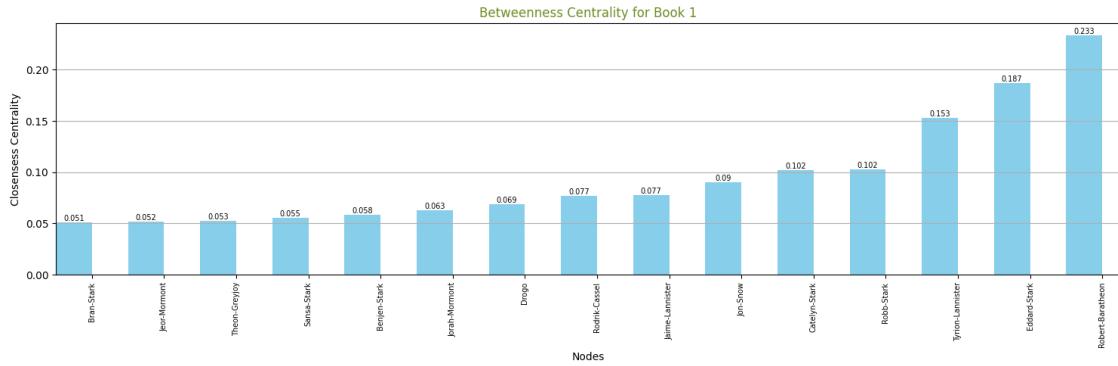
```
[ ]: bet_1 = nx.betweenness_centrality(G_1, weight='weight')
bet_2 = nx.betweenness_centrality(G_2, weight='weight')
bet_3 = nx.betweenness_centrality(G_3, weight='weight')
```

```

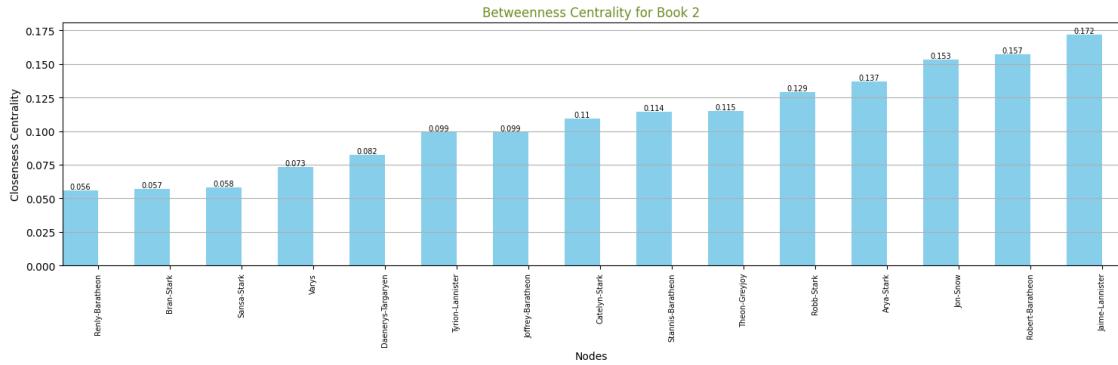
bet_4 = nx.betweenness_centrality(G_4, weight='weight')
bet_5 = nx.betweenness_centrality(G_5, weight='weight')
bet_all = nx.betweenness_centrality(G_all, weight='weight')

```

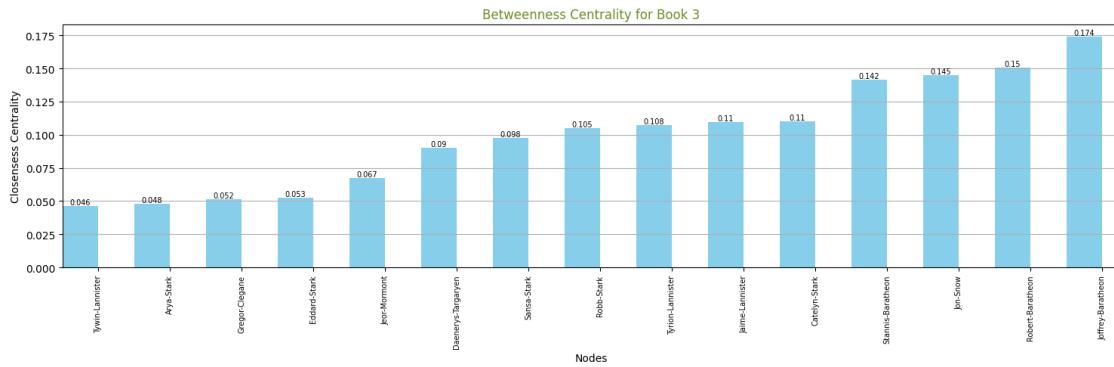
[]: top_bet_nodes_1 = plot_betweenness_centrality(bet_1, 'Betweenness Centrality for Book 1', font_size=7, top_n=15)



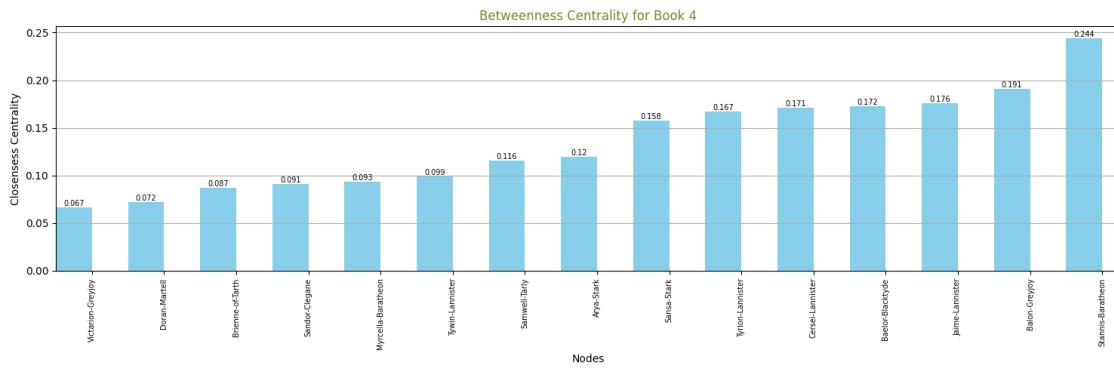
[]: top_bet_nodes_2 = plot_betweenness_centrality(bet_2, 'Betweenness Centrality for Book 2', font_size=7, top_n=15)



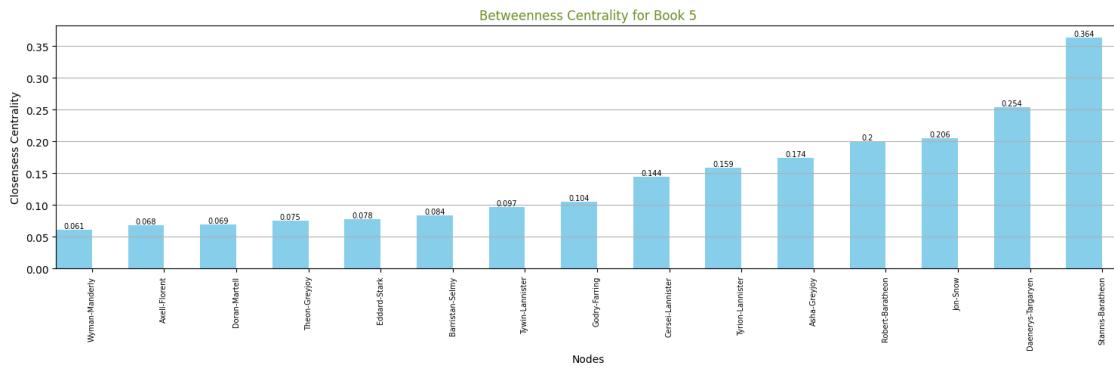
[]: top_bet_nodes_3 = plot_betweenness_centrality(bet_3, 'Betweenness Centrality for Book 3', font_size=7, top_n=15)



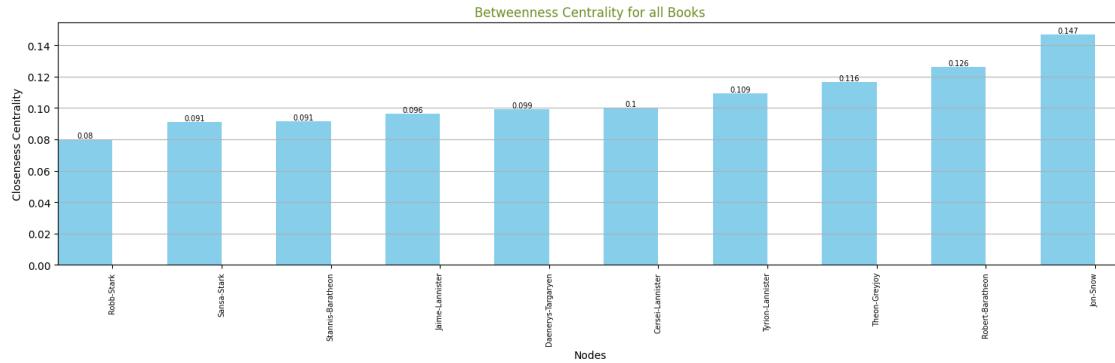
```
[ ]: top_bet_nodes_4 = plot_betweenness_centrality(bet_4, 'Betweenness Centrality for Book 4', font_size=7, top_n=15)
```



```
[ ]: top_bet_nodes_5 = plot_betweenness_centrality(bet_5, 'Betweenness Centrality for Book 5', font_size=7, top_n=15)
```



```
[ ]: top_bet_nodes_all = plot_betweenness_centrality(bet_all, 'Betweenness Centrality for all Books', font_size=7, top_n=10)
```

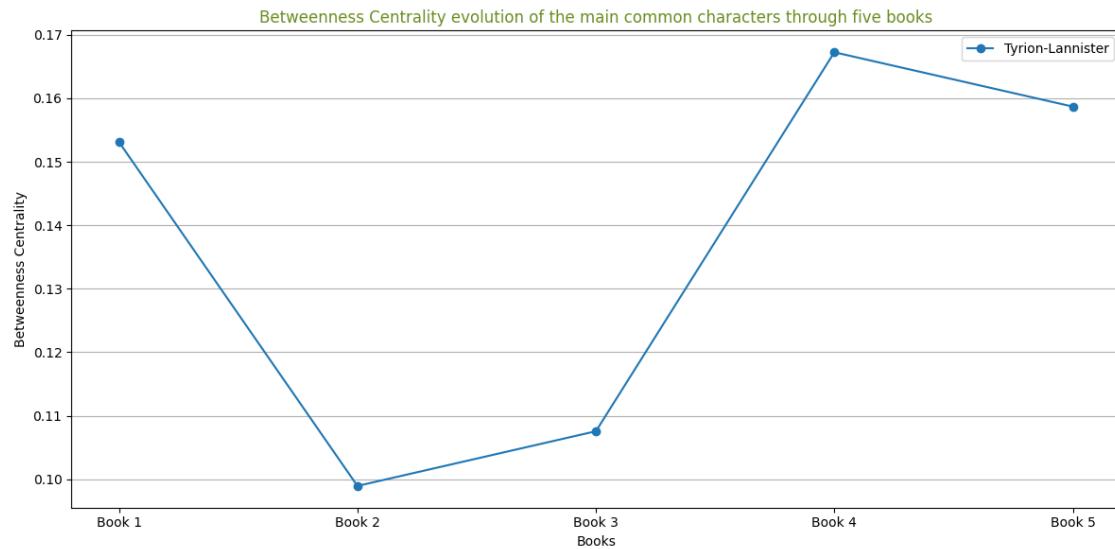


```
[ ]: top_common_nodes_BC = common_keys([top_bet_nodes_1, top_bet_nodes_2, top_bet_nodes_3, top_bet_nodes_4, top_bet_nodes_5])
```

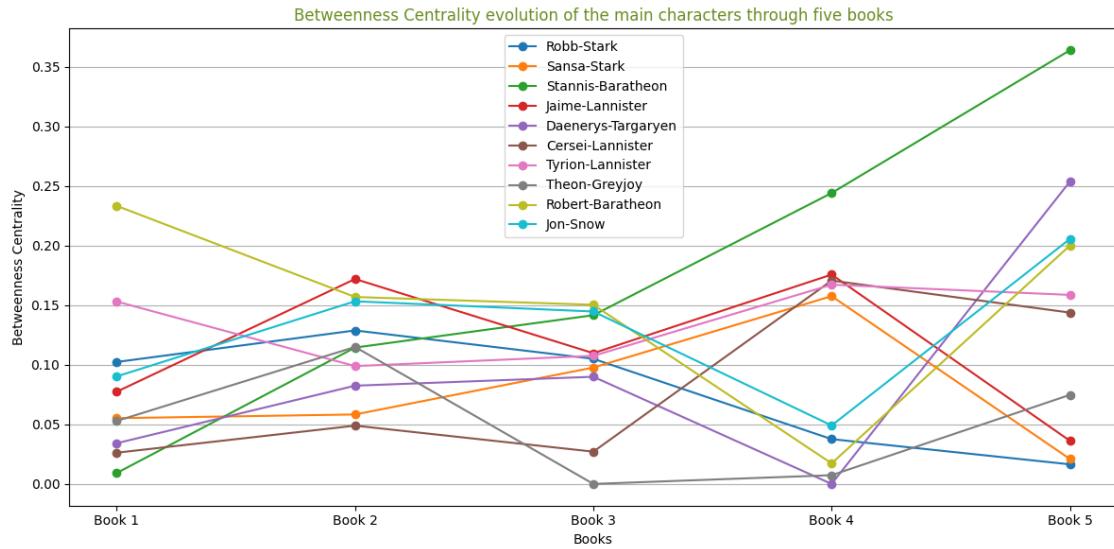
```
[ ]: top_common_nodes_BC
```

```
[ ]: ['Tyrion-Lannister']
```

```
[ ]: show_evolution([top_bet_nodes_1, top_bet_nodes_2, top_bet_nodes_3, top_bet_nodes_4, top_bet_nodes_5], top_common_nodes_BC, title="Betweenness Centrality evolution of the main common characters through five books", xlabel="Books", ylabel="Betweenness Centrality")
```



```
[ ]: show_evolution([bet_1, bet_2, bet_3, bet_4, bet_5],
                  top_bet_nodes_all,
                  title="Betweenness Centrality evolution of the main
                  ↵characters through five books",
                  xlabel="Books",
                  ylabel="Betweenness Centrality")
```



6.1.6 Clusters of Nodes

Cliques

```
[ ]: def find_family_cliques(cliques, family_names):
    """
    Trova i cliques in cui i nomi della famiglia rappresentano la maggior parte
    ↵dei membri.

    :param cliques: Lista di cliques, dove ogni clique è una lista di nomi.
    :param family_names: Lista di stringhe contenenti parte dei nomi dei
    ↵personaggi.

    :return: Lista di cliques in cui i membri corrispondono alla maggior parte
    ↵della famiglia.
    """
    family_cliques = []

    # Convertire la lista di nomi della famiglia in un set per un confronto più
    ↵veloce
```

```

family_set = set(family_names)

for clique in cliques:
    # Contare quanti membri del clique corrispondono ai nomi della famiglia
    count = sum(1 for member in clique if any(name in member for name in family_set))

        # Verificare se la maggior parte dei membri del clique appartiene alla famiglia
    if count >= (3 * len(clique)) / 5:
        family_cliques.append(clique)

return family_cliques

```

```

[ ]: def find_largest_cliques(cliques, top_n=5):
    """
    Trova i 'top_n' cliques più grandi da una lista di cliques.

    :param cliques: Lista di cliques, dove ogni clique è una lista di membri.
    :param top_n: Numero di cliques più grandi da restituire.
    :return: Lista dei 'top_n' cliques più grandi.
    """

    # Ordinare i cliques in base alla loro dimensione (lunghezza)
    sorted_cliques = sorted(cliques, key=len, reverse=True)

    # Restituire i 'top_n' cliques più grandi
    return sorted_cliques[:top_n]

```

```

[ ]: def find_common_cliques(cliques_list):
    """
    Trova i cliques comuni in una lista di liste di cliques.

    :param cliques_list: Lista di liste di cliques.
    :return: Lista di cliques comuni.
    """

    if not cliques_list:
        return []

    # Inizializzare il set dei cliques comuni con i cliques della prima lista
    common_cliques = set(map(tuple, cliques_list[0]))

    # Iterare attraverso le altre liste di cliques
    for cliques in cliques_list[1:]:
        common_cliques.intersection_update(set(map(tuple, cliques)))

    # Convertire i risultati in una lista di liste
    return [list(clique) for clique in common_cliques]

```

```
[ ]: cliques_1 = list(nx.find_cliques(G_1))
cliques_2 = list(nx.find_cliques(G_2))
cliques_3 = list(nx.find_cliques(G_3))
cliques_4 = list(nx.find_cliques(G_4))
cliques_5 = list(nx.find_cliques(G_5))
cliques_all = list(nx.find_cliques(G_all))
```

```
[ ]: # Stampo i cliques più grandi, sembrano identificare i gruppi di personaggi che ↴ nella storia interagiscono tra loro
for clique in cliques_1:
    if len(clique) > 9:
        print(clique)
```

```
['Eddard-Stark', 'Robert-Baratheon', 'Sansa-Stark', 'Joffrey-Baratheon',
'Cersei-Lannister', 'Petyr-Baelish', 'Catelyn-Stark', 'Tyrion-Lannister',
'Varys', 'Jaime-Lannister']
['Eddard-Stark', 'Robert-Baratheon', 'Sansa-Stark', 'Joffrey-Baratheon',
'Cersei-Lannister', 'Petyr-Baelish', 'Renly-Baratheon', 'Pycelle', 'Varys',
'Barristan-Selmy']
['Eddard-Stark', 'Robert-Baratheon', 'Sansa-Stark', 'Joffrey-Baratheon',
'Cersei-Lannister', 'Petyr-Baelish', 'Renly-Baratheon', 'Jaime-Lannister',
'Barristan-Selmy', 'Varys']
['Eddard-Stark', 'Robert-Baratheon', 'Sansa-Stark', 'Joffrey-Baratheon',
'Cersei-Lannister', 'Petyr-Baelish', 'Renly-Baratheon', 'Jaime-Lannister',
'Barristan-Selmy', 'Sandor-Clegane']
['Eddard-Stark', 'Robert-Baratheon', 'Sansa-Stark', 'Joffrey-Baratheon',
'Cersei-Lannister', 'Petyr-Baelish', 'Renly-Baratheon', 'Jaime-Lannister',
'Tyrion-Lannister', 'Varys']
['Eddard-Stark', 'Robert-Baratheon', 'Sansa-Stark', 'Joffrey-Baratheon',
'Cersei-Lannister', 'Petyr-Baelish', 'Renly-Baratheon', 'Jaime-Lannister',
'Tyrion-Lannister', 'Sandor-Clegane']
```

```
[ ]: # Molti cliques nel libro uno contengono gli Stark perchè buona parte della ↴ storia gira intorno a loro. Quindi stampo i cliques che rappresentano la ↴ loro famiglia
Stark_family_clique = find_family_cliques(cliques_1, family_names=["Stark"])
for clique in Stark_family_clique:
    if len(clique) > 4:
        print(clique)
```

```
['Eddard-Stark', 'Rickon-Stark', 'Robb-Stark', 'Catelyn-Stark', 'Jon-Snow',
'Bran-Stark', 'Sansa-Stark', 'Tyrion-Lannister']
['Eddard-Stark', 'Rickon-Stark', 'Robb-Stark', 'Catelyn-Stark', 'Jon-Snow',
'Bran-Stark', 'Sansa-Stark', 'Arya-Stark']
['Eddard-Stark', 'Rickon-Stark', 'Robb-Stark', 'Catelyn-Stark', 'Jon-Snow',
'Bran-Stark', 'Luwin', 'Tyrion-Lannister']
['Eddard-Stark', 'Luwin', 'Robb-Stark', 'Rickard-Karstark', 'Catelyn-Stark'],
```

```

'Bran-Stark']
['Eddard-Stark', 'Edmure-Tully', 'Jaime-Lannister', 'Catelyn-Stark', 'Robb-
Stark']
['Eddard-Stark', 'Rodrik-Cassel', 'Joffrey-Baratheon', 'Catelyn-Stark', 'Robb-
Stark', 'Jon-Snow', 'Bran-Stark', 'Arya-Stark']
['Eddard-Stark', 'Rickard-Karstark', 'Robb-Stark', 'Catelyn-Stark', 'Jaime-
Lannister']
['Eddard-Stark', 'Robert-Baratheon', 'Sansa-Stark', 'Joffrey-Baratheon', 'Robb-
Stark', 'Catelyn-Stark', 'Jon-Snow', 'Bran-Stark', 'Arya-Stark']
['Eddard-Stark', 'Robert-Baratheon', 'Sansa-Stark', 'Jory-Cassel', 'Robb-Stark',
'Jon-Snow', 'Bran-Stark', 'Arya-Stark']
['Yoren', 'Benjen-Stark', 'Robb-Stark', 'Jon-Snow', 'Bran-Stark', 'Arya-Stark']
['Benjen-Stark', 'Jon-Snow', 'Bran-Stark', 'Robb-Stark', 'Arya-Stark', 'Robert-
Baratheon']

```

```
[ ]: Targaryen_family_clique = find_family_cliques(cliques_1,
    ↪family_names=["Targaryen", "Mormont"])
for clique in Targaryen_family_clique:
    if len(clique) > 2:
        print(clique)
```

```

['Eddard-Stark', 'Aegon-I-Targaryen', 'Daenerys-Targaryen']
['Viserys-Targaryen', 'Daenerys-Targaryen', 'Jorah-Mormont', 'Rhaegar-
Targaryen', 'Robert-Baratheon']
```

```
[ ]: # Nel libro due molti cliques contengono i cognomi Lannister e Baratheon,
    ↪proprio perché buona parte della storia è incentrata su di loro
LannisterBaratheon_family_clique = find_family_cliques(cliques_2,
    ↪family_names=["Lannister", "Baratheon"])
for clique in LannisterBaratheon_family_clique:
    if len(clique) > 6:
        print(clique)
```

```

['Tyrion-Lannister', 'Joffrey-Baratheon', 'Cersei-Lannister', 'Sansa-Stark',
'Jaime-Lannister', 'Stannis-Baratheon', 'Renly-Baratheon', 'Robb-Stark',
'Catelyn-Stark', 'Robert-Baratheon']
['Tyrion-Lannister', 'Joffrey-Baratheon', 'Cersei-Lannister', 'Sansa-Stark',
'Jaime-Lannister', 'Stannis-Baratheon', 'Renly-Baratheon', 'Petyr-Baelish']
['Tyrion-Lannister', 'Joffrey-Baratheon', 'Cersei-Lannister', 'Sansa-Stark',
'Jaime-Lannister', 'Arya-Stark', 'Robb-Stark', 'Robert-Baratheon']
['Tyrion-Lannister', 'Joffrey-Baratheon', 'Cersei-Lannister', 'Sansa-Stark',
'Myrcella-Baratheon', 'Sandor-Clegane', 'Tommen-Baratheon']
['Tyrion-Lannister', 'Joffrey-Baratheon', 'Cersei-Lannister', 'Eddard-Stark',
```

```
'Stannis-Baratheon', 'Renly-Baratheon', 'Robb-Stark', 'Robert-Baratheon',
'Catelyn-Stark']
[['Tyrion-Lannister', 'Joffrey-Baratheon', 'Cersei-Lannister', 'Eddard-Stark',
'Stannis-Baratheon', 'Renly-Baratheon', 'Robb-Stark', 'Robert-Baratheon',
'Varys']]
[['Tyrion-Lannister', 'Joffrey-Baratheon', 'Cersei-Lannister', 'Eddard-Stark',
'Stannis-Baratheon', 'Renly-Baratheon', 'Petyr-Baelish', 'Varys']]
[['Tyrion-Lannister', 'Joffrey-Baratheon', 'Cersei-Lannister', 'Pycelle',
'Stannis-Baratheon', 'Varys', 'Robert-Baratheon']]
[['Tyrion-Lannister', 'Joffrey-Baratheon', 'Cersei-Lannister', 'Tywin-Lannister',
'Jaime-Lannister', 'Robb-Stark', 'Robert-Baratheon', 'Catelyn-Stark', 'Stannis-
Baratheon', 'Renly-Baratheon']]
[['Tyrion-Lannister', 'Joffrey-Baratheon', 'Cersei-Lannister', 'Tywin-Lannister',
'Jaime-Lannister', 'Robb-Stark', 'Robert-Baratheon', 'Arya-Stark']]
[['Tyrion-Lannister', 'Joffrey-Baratheon', 'Loras-Tyrell', 'Catelyn-Stark',
'Stannis-Baratheon', 'Robert-Baratheon', 'Renly-Baratheon']]
```

```
[ ]: Baratheon_family_clique = find_family_cliques(cliques_2,
    ↪family_names=["Baratheon"])
for clique in Baratheon_family_clique:
    if len(clique) > 1:
        print(clique)
```

```
['Steffon-Baratheon', 'Stannis-Baratheon', 'Renly-Baratheon']
['Mace-Tyrell', 'Stannis-Baratheon', 'Renly-Baratheon', 'Loras-Tyrell',
'Joffrey-Baratheon']
['Cressen', 'Stannis-Baratheon', 'Robert-Baratheon', 'Renly-Baratheon']
```

```
[ ]: # Stampo i cliques più grandi, sembrano identificare i gruppi di presonaggi che
    ↪nella storia interagiscono tra loro
for clique in cliques_2:
    if len(clique) > 8:
        print(clique)
```

```
['Amory-Lorch', 'Joffrey-Baratheon', 'Cersei-Lannister', 'Gregor-Clegane',
'Polliver', 'Chiswyck', 'Rafford', 'Dunsen', 'Meryn-Trant', 'Sandor-Clegane',
'Ilyn-Payne', 'Tickler']
[['Amory-Lorch', 'Joffrey-Baratheon', 'Cersei-Lannister', 'Gregor-Clegane',
'Polliver', 'Chiswyck', 'Rafford', 'Dunsen', 'Arya-Stark']]
[['Tyrion-Lannister', 'Joffrey-Baratheon', 'Cersei-Lannister', 'Sansa-Stark',
'Jaime-Lannister', 'Stannis-Baratheon', 'Renly-Baratheon', 'Robb-Stark',
'Catelyn-Stark', 'Robert-Baratheon']]
[['Tyrion-Lannister', 'Joffrey-Baratheon', 'Cersei-Lannister', 'Eddard-Stark',
```

```
'Stannis-Baratheon', 'Renly-Baratheon', 'Robb-Stark', 'Robert-Baratheon',
'Catelyn-Stark']
['Tyrion-Lannister', 'Joffrey-Baratheon', 'Cersei-Lannister', 'Eddard-Stark',
'Stannis-Baratheon', 'Renly-Baratheon', 'Robb-Stark', 'Robert-Baratheon',
'Varys']
['Tyrion-Lannister', 'Joffrey-Baratheon', 'Cersei-Lannister', 'Tywin-Lannister',
'Jaime-Lannister', 'Robb-Stark', 'Robert-Baratheon', 'Catelyn-Stark', 'Stannis-
Baratheon', 'Renly-Baratheon']
```

[]: # Oberyn-Martell death

```
Oberyn_Martell_death_clique = find_family_cliques(cliques_3,
    ↪family_names=["Martell", "Lannister"])
for clique in Oberyn_Martell_death_clique:
    if len(clique) > 5:
        print(clique)
```

```
['Tyrion-Lannister', 'Joffrey-Baratheon', 'Sansa-Stark', 'Cersei-Lannister',
'Jaime-Lannister', 'Tywin-Lannister', 'Oberyn-Martell']
['Tyrion-Lannister', 'Joffrey-Baratheon', 'Gregor-Clegane', 'Cersei-Lannister',
'Jaime-Lannister', 'Oberyn-Martell', 'Tywin-Lannister']
['Tyrion-Lannister', 'Elia-Martell', 'Gregor-Clegane', 'Cersei-Lannister',
'Oberyn-Martell', 'Jaime-Lannister']
```

[]: Frey_family_clique = find_family_cliques(cliques_3, family_names=["Frey"])

```
for clique in Frey_family_clique:
    if len(clique) > 1:
        print(clique)
```

```
['Walder-Frey', 'Petyr-Frey']
['Ryman-Frey', 'Robb-Stark', 'Olyvar-Frey']
['Ryman-Frey', 'Robb-Stark', 'Stevron-Frey']
['Ryman-Frey', 'Petyr-Frey']
['Merrett-Frey', 'Lothar-Frey']
['Stevron-Frey', 'Robb-Stark', 'Aegon-Frey-(son-of-Stevron)']
```

[]: Tyrell_family_clique = find_family_cliques(cliques_3, family_names=["Tyrell"])

```
for clique in Tyrell_family_clique:
    if len(clique) > 1:
        print(clique)
```

```
['Alerie-Hightower', 'Margaery-Tyrell', 'Mace-Tyrell']
['Elinor-Tyrell', 'Alla-Tyrell', 'Megga-Tyrell', 'Sansa-Stark']
['Tyrion-Lannister', 'Willas-Tyrell', 'Sansa-Stark', 'Garlan-Tyrell', 'Loras-
```

```
Tyrell']
['Tyrion-Lannister', 'Willas-Tyrell', 'Sansa-Stark', 'Margaery-Tyrell', 'Loras-Tyrell']
```

```
[ ]: Greyjoy_family_clique = find_family_cliques(cliques_4, family_names=["Greyjoy"])
for clique in Greyjoy_family_clique:
    if len(clique) > 3:
        print(clique)
```

```
['Asha-Greyjoy', 'Euron-Greyjoy', 'Victarion-Greyjoy', 'Aeron-Greyjoy', 'Balon-Greyjoy']
['Asha-Greyjoy', 'Euron-Greyjoy', 'Victarion-Greyjoy', 'Rodrik-Harlaw']
['Baelor-Blacktyde', 'Aeron-Greyjoy', 'Balon-Greyjoy', 'Euron-Greyjoy',
'Victarion-Greyjoy']
```

```
[ ]: common_cliques = find_common_cliques([cliques_1, cliques_2, cliques_3,
cliques_4, cliques_5])
if common_cliques:
    print("Common found Cliques:")
    for clique in common_cliques:
        print(clique)
else:
    print("None Common Clique found.")
```

None Common Clique found.

```
[ ]: top_cliques_all = find_largest_cliques(cliques_all, top_n=15)
```

```
[ ]: print("The greatest cliques among all books are:")
for clique in top_cliques_all:
    print(clique)
```

The greatest cliques among all books are:

```
['Tyrion-Lannister', 'Cersei-Lannister', 'Joffrey-Baratheon', 'Sansa-Stark',
'Jaime-Lannister', 'Robert-Baratheon', 'Eddard-Stark', 'Petyr-Baelish', 'Renly-Baratheon',
'Stannis-Baratheon', 'Tywin-Lannister', 'Varys', 'Robb-Stark',
'Catelyn-Stark']
['Dunsen', 'Polliver', 'Chiswyck', 'Gregor-Clegane', 'Arya-Stark', 'Amory-Lorch',
'Cersei-Lannister', 'Joffrey-Baratheon', 'Meryn-Trant', 'Sandor-Clegane',
'Ilyn-Payne', 'Rafford', 'Tickler']
['Tyrion-Lannister', 'Cersei-Lannister', 'Joffrey-Baratheon', 'Sansa-Stark',
'Jaime-Lannister', 'Robert-Baratheon', 'Eddard-Stark', 'Petyr-Baelish', 'Renly-Baratheon',
'Stannis-Baratheon', 'Tywin-Lannister', 'Varys', 'Pycelle']
```

```

['Tyrion-Lannister', 'Cersei-Lannister', 'Joffrey-Baratheon', 'Sansa-Stark',
'Jaime-Lannister', 'Robert-Baratheon', 'Eddard-Stark', 'Petyr-Baelish', 'Gregor-
Clegane', 'Tywin-Lannister', 'Robb-Stark', 'Arya-Stark']
['Tyrion-Lannister', 'Cersei-Lannister', 'Joffrey-Baratheon', 'Sansa-Stark',
'Jaime-Lannister', 'Robert-Baratheon', 'Eddard-Stark', 'Petyr-Baelish', 'Renly-
Baratheon', 'Stannis-Baratheon', 'Loras-Tyrell', 'Catelyn-Stark']
['Tyrion-Lannister', 'Cersei-Lannister', 'Joffrey-Baratheon', 'Sansa-Stark',
'Jaime-Lannister', 'Robert-Baratheon', 'Eddard-Stark', 'Petyr-Baelish', 'Renly-
Baratheon', 'Barristan-Selmy', 'Varys', 'Pycelle']
['Tyrion-Lannister', 'Cersei-Lannister', 'Joffrey-Baratheon', 'Sansa-Stark',
'Jaime-Lannister', 'Robert-Baratheon', 'Eddard-Stark', 'Petyr-Baelish', 'Arya-
Stark', 'Catelyn-Stark', 'Robb-Stark', 'Tywin-Lannister']
['Tyrion-Lannister', 'Cersei-Lannister', 'Joffrey-Baratheon', 'Sansa-Stark',
'Jaime-Lannister', 'Robert-Baratheon', 'Brienne-of-Tarth', 'Renly-Baratheon',
'Stannis-Baratheon', 'Catelyn-Stark', 'Robb-Stark', 'Tywin-Lannister']
['Tyrion-Lannister', 'Cersei-Lannister', 'Joffrey-Baratheon', 'Sansa-Stark',
'Jaime-Lannister', 'Robert-Baratheon', 'Eddard-Stark', 'Petyr-Baelish', 'Gregor-
Clegane', 'Loras-Tyrell', 'Sandor-Clegane']
['Tyrion-Lannister', 'Cersei-Lannister', 'Joffrey-Baratheon', 'Sansa-Stark',
'Jaime-Lannister', 'Robert-Baratheon', 'Eddard-Stark', 'Petyr-Baelish', 'Gregor-
Clegane', 'Sandor-Clegane', 'Arya-Stark']
['Tyrion-Lannister', 'Cersei-Lannister', 'Joffrey-Baratheon', 'Sansa-Stark',
'Jaime-Lannister', 'Robert-Baratheon', 'Eddard-Stark', 'Petyr-Baelish', 'Gregor-
Clegane', 'Tywin-Lannister', 'Pycelle']
['Tyrion-Lannister', 'Cersei-Lannister', 'Joffrey-Baratheon', 'Sansa-Stark',
'Jaime-Lannister', 'Robert-Baratheon', 'Eddard-Stark', 'Petyr-Baelish', 'Renly-
Baratheon', 'Barristan-Selmy', 'Sandor-Clegane']
['Tyrion-Lannister', 'Cersei-Lannister', 'Joffrey-Baratheon', 'Sansa-Stark',
'Jaime-Lannister', 'Robert-Baratheon', 'Eddard-Stark', 'Petyr-Baelish', 'Renly-
Baratheon', 'Sandor-Clegane', 'Loras-Tyrell']
['Tyrion-Lannister', 'Cersei-Lannister', 'Joffrey-Baratheon', 'Sansa-Stark',
'Jaime-Lannister', 'Robert-Baratheon', 'Eddard-Stark', 'Jon-Snow', 'Robb-Stark',
'Catelyn-Stark', 'Stannis-Baratheon']
['Tyrion-Lannister', 'Cersei-Lannister', 'Joffrey-Baratheon', 'Sansa-Stark',
'Jaime-Lannister', 'Robert-Baratheon', 'Eddard-Stark', 'Jon-Snow', 'Robb-Stark',
'Catelyn-Stark', 'Arya-Stark']

```

Cores I K-Cores sono una misura di centralità, servono spesso per vedere quali strutture sono centrali nel grafo e quali sono periferiche. Quindi ha senso cercare i core più grandi in un grafo.

Ha anche senso vedere se ci sono dei personaggi presenti nei 5 libri nei core più grandi. In teoria, se un libro si concentra su un certo gruppo di personaggi allora il core con k grande dovrebbe avere dentro quei personaggi.

```
[ ]: def plot_k_core(k_core, title, font_size=10):

    plt.figure(figsize=(12, 6))
```

```

    pos = nx.spring_layout(k_core)
    nx.draw(k_core, pos, with_labels=True, node_size=500,
    ↵node_color='lightblue', font_size=font_size, font_color='red')

    # Mostrare il grafo
    plt.title(title, fontsize=18, color="blue")
    plt.show()

    return list(k_core.nodes)

```

```

[ ]: # k values to compute the K-Cores
k = [2,3,4,5,6,7,8,9,10,11,12,13,14]

k_core_1 = []
k_core_2 = []
k_core_3 = []
k_core_4 = []
k_core_5 = []
k_core_all = []

for i in k:
    k_core_1[i] = list(nx.k_core(G_1, k=i, core_number=None).nodes)
    k_core_2[i] = list(nx.k_core(G_2, k=i, core_number=None).nodes)
    k_core_3[i] = list(nx.k_core(G_3, k=i, core_number=None).nodes)
    k_core_4[i] = list(nx.k_core(G_4, k=i, core_number=None).nodes)
    k_core_5[i] = list(nx.k_core(G_5, k=i, core_number=None).nodes)
    k_core_all[i] = list(nx.k_core(G_all, k=i, core_number=None).nodes)

print("All found K-Cores in Book 1")
for key, value in k_core_1.items():
    print(f"{key}-Core: {value}")

print("\nAll found K-Cores in Book 2")
for key, value in k_core_2.items():
    print(f"{key}-Core: {value}")

print("\nAll found K-Cores in Book 3")
for key, value in k_core_3.items():
    print(f"{key}-Core: {value}")

print("\nAll found K-Cores in Book 4")
for key, value in k_core_4.items():

```

```

print(f"{key}-Core: {value}")

print("\nAll found K-Cores in Book 5")
for key, value in k_core_5.items():
    print(f"{key}-Core: {value}")

print("\nAll found K-Cores in All Books")
for key, value in k_core_all.items():
    print(f"{key}-Core: {value}")

```

All found K-Cores in Book 1

2-Core: ['Addam-Marbrand', 'Jaime-Lannister', 'Tywin-Lannister', 'Aegon-I-Targaryen', 'Daenerys-Targaryen', 'Eddard-Stark', 'Aemon-Targaryen-(Maester-Aemon)', 'Alliser-Thorne', 'Bowen-Marsh', 'Chett', 'Jeor-Mormont', 'Jon-Snow', 'Samwell-Tarly', 'Aerys-II-Targaryen', 'Brandon-Stark', 'Gerold-Hightower', 'Jon-Arryn', 'Robert-Baratheon', 'Aggo', 'Drogo', 'Jhogo', 'Jorah-Mormont', 'Quaro', 'Rakharo', 'Albett', 'Halder', 'Rast', 'Grenn', 'Pypar', 'Tyrion-Lannister', 'Alyn', 'Harwin', 'Jory-Cassel', 'Tomard', 'Arya-Stark', 'Benjen-Stark', 'Bran-Stark', 'Catelyn-Stark', 'Cersei-Lannister', 'Desmond', 'Ilyn-Payne', 'Jeyne-Poole', 'Joffrey-Baratheon', 'Meryn-Trant', 'Mordane', 'Mycah', 'Myrcella-Baratheon', 'Petyr-Baelish', 'Rickon-Stark', 'Robb-Stark', 'Rodrik-Cassel', 'Sandor-Clegane', 'Sansa-Stark', 'Syrio-Forel', 'Tommen-Baratheon', 'Vayon-Poole', 'Yoren', 'Balon-Greyjoy', 'Balon-Swann', 'Renly-Baratheon', 'Barristan-Selmy', 'Boros-Blount', 'Pycelle', 'Varys', 'Jaremy-Rykker', 'Luwin', 'Theon-Greyjoy', 'Waymar-Royce', 'Beric-Dondarrion', 'Gregor-Clegane', 'Loras-Tyrell', 'Hallis-Mollen', 'Hodor', 'Hullen', 'Josphere', 'Nan', 'Osha', 'Rickard-Karstark', 'Rickard-Stark', 'Stiv', 'Lyanna-Stark', 'Bronn', 'Chiggen', 'Marillion', 'Shae', 'Shagga', 'Vardis-Egen', 'Willis-Wode', 'Brynden-Tully', 'Edmure-Tully', 'Hoster-Tully', 'Lysa-Arryn', 'Nestor-Royce', 'Walder-Frey', 'Colemon', 'Eon-Hunter', 'Jon-Umber-(Greatjon)', 'Robert-Arryn', 'Stevron-Frey', 'Tytos-Blackwood', 'Wendel-Manderly', 'Cayn', 'Janos-Slynt', 'Stannis-Baratheon', 'Karyl-Vance', 'Cohollo', 'Haggo', 'Qotho', 'Conn', 'Doreah', 'Illyrio-Mopatis', 'Irri', 'Jhiqui', 'Mirri-Maz-Duur', 'Rhaegar-Targaryen', 'Viserys-Targaryen', 'Danwell-Frey', 'Hosteen-Frey', 'Jared-Frey', 'Daryn-Hornwood', 'Torrhen-Karstark', 'Jommo', 'Ogo', 'Galbart-Glover', 'High-Septon-(fat_one)', 'Marq-Piper', 'Raymun-Darry', 'Wyl-(guard)', 'Roose-Bolton', 'Gared', 'Will-(prologue)', 'Todder', 'Heward', 'Kevan-Lannister', 'Maege-Mormont', 'Jyck', 'Morrec', 'Timett']

3-Core: ['Jaime-Lannister', 'Tywin-Lannister', 'Daenerys-Targaryen', 'Eddard-Stark', 'Aemon-Targaryen-(Maester-Aemon)', 'Alliser-Thorne', 'Bowen-Marsh', 'Chett', 'Jeor-Mormont', 'Jon-Snow', 'Samwell-Tarly', 'Aerys-II-Targaryen', 'Brandon-Stark', 'Gerold-Hightower', 'Jon-Arryn', 'Robert-Baratheon', 'Aggo', 'Drogo', 'Jhogo', 'Jorah-Mormont', 'Quaro', 'Rakharo', 'Albett', 'Halder', 'Rast', 'Grenn', 'Pypar', 'Tyrion-Lannister', 'Alyn', 'Harwin', 'Jory-Cassel', 'Tomard', 'Arya-Stark', 'Benjen-Stark', 'Bran-Stark', 'Catelyn-Stark', 'Cersei-

Lannister', 'Ilyn-Payne', 'Jeyne-Poole', 'Joffrey-Baratheon', 'Meryn-Trant', 'Mordane', 'Mycah', 'Myrcella-Baratheon', 'Petyr-Baelish', 'Rickon-Stark', 'Robb-Stark', 'Rodrik-Cassel', 'Sandor-Clegane', 'Sansa-Stark', 'Syrio-Forel', 'Tommen-Baratheon', 'Vayon-Poole', 'Yoren', 'Renly-Baratheon', 'Barristan-Selmy', 'Boros-Blount', 'Pycelle', 'Varys', 'Jaremy-Rykker', 'Luwin', 'Theon-Greyjoy', 'Beric-Dondarrion', 'Gregor-Clegane', 'Loras-Tyrell', 'Hallis-Mollen', 'Hodor', 'Hullen', 'Nan', 'Osha', 'Rickard-Karstark', 'Lyanna-Stark', 'Bronn', 'Chiggen', 'Marillion', 'Vardis-Egen', 'Willis-Wode', 'Brynden-Tully', 'Edmure-Tully', 'Hoster-Tully', 'Lysa-Arryn', 'Walder-Frey', 'Eon-Hunter', 'Jon-Umber-(Greatjon)', 'Robert-Arryn', 'Cayn', 'Janos-Slynt', 'Stannis-Baratheon', 'Cohollo', 'Haggo', 'Qotho', 'Doreah', 'Illyrio-Mopatis', 'Irri', 'Jhiqui', 'Mirri-Maz-Duur', 'Rhaegar-Targaryen', 'Viserys-Targaryen', 'Galbart-Glover', 'Marq-Piper', 'Wyl-(guard)', 'Roose-Bolton', 'Todder', 'Kevan-Lannister']
4-Core: ['Beric-Dondarrion', 'Marq-Piper', 'Rickon-Stark', 'Hullen', 'Drogo', 'Luwin', 'Hallis-Mollen', 'Walder-Frey', 'Jeor-Mormont', 'Irri', 'Barristan-Selmy', 'Lyanna-Stark', 'Rhaegar-Targaryen', 'Tommen-Baratheon', 'Eddard-Stark', 'Haggo', 'Meryn-Trant', 'Galbart-Glover', 'Rakharo', 'Alyn', 'Cersei-Lannister', 'Nan', 'Rickard-Karstark', 'Robb-Stark', 'Tomard', 'Bowen-Marsh', 'Viserys-Targaryen', 'Joffrey-Baratheon', 'Bronn', 'Halder', 'Varys', 'Jhiqui', 'Willis-Wode', 'Edmure-Tully', 'Aemon-Targaryen-(Maester-Aemon)', 'Brynden-Tully', 'Pypar', 'Rast', 'Jon-Snow', 'Roose-Bolton', 'Daenerys-Targaryen', 'Alliser-Thorne', 'Jhogo', 'Arya-Stark', 'Boros-Blount', 'Loras-Tyrell', 'Todder', 'Ilyn-Payne', 'Marillion', 'Tywin-Lannister', 'Pycelle', 'Doreah', 'Yoren', 'Theon-Greyjoy', 'Bran-Stark', 'Petyr-Baelish', 'Mirri-Maz-Duur', 'Renly-Baratheon', 'Sansa-Stark', 'Benjen-Stark', 'Myrcella-Baratheon', 'Grenn', 'Vardis-Egen', 'Aerys-II-Targaryen', 'Rodrik-Cassel', 'Gregor-Clegane', 'Aggo', 'Illyrio-Mopatis', 'Hodor', 'Hoster-Tully', 'Jon-Arryn', 'Catelyn-Stark', 'Samwell-Tarly', 'Tyrion-Lannister', 'Cohollo', 'Qotho', 'Sandor-Clegane', 'Mordane', 'Brandon-Stark', 'Lysa-Arryn', 'Jon-Umber-(Greatjon)', 'Vayon-Poole', 'Jorah-Mormont', 'Mycah', 'Jory-Cassel', 'Jeyne-Poole', 'Quaro', 'Stannis-Baratheon', 'Jaime-Lannister', 'Harwin', 'Robert-Baratheon', 'Janos-Slynt']
5-Core: ['Beric-Dondarrion', 'Rickon-Stark', 'Hullen', 'Drogo', 'Luwin', 'Hallis-Mollen', 'Walder-Frey', 'Jeor-Mormont', 'Irri', 'Barristan-Selmy', 'Rhaegar-Targaryen', 'Tommen-Baratheon', 'Eddard-Stark', 'Haggo', 'Meryn-Trant', 'Galbart-Glover', 'Rakharo', 'Cersei-Lannister', 'Rickard-Karstark', 'Robb-Stark', 'Tomard', 'Bowen-Marsh', 'Viserys-Targaryen', 'Joffrey-Baratheon', 'Bronn', 'Halder', 'Varys', 'Jhiqui', 'Willis-Wode', 'Edmure-Tully', 'Aemon-Targaryen-(Maester-Aemon)', 'Brynden-Tully', 'Pypar', 'Rast', 'Jon-Snow', 'Roose-Bolton', 'Daenerys-Targaryen', 'Alliser-Thorne', 'Jhogo', 'Arya-Stark', 'Boros-Blount', 'Loras-Tyrell', 'Ilyn-Payne', 'Marillion', 'Tywin-Lannister', 'Pycelle', 'Doreah', 'Yoren', 'Theon-Greyjoy', 'Bran-Stark', 'Petyr-Baelish', 'Mirri-Maz-Duur', 'Renly-Baratheon', 'Sansa-Stark', 'Benjen-Stark', 'Myrcella-Baratheon', 'Grenn', 'Rodrik-Cassel', 'Gregor-Clegane', 'Aggo', 'Illyrio-Mopatis', 'Hoster-Tully', 'Jon-Arryn', 'Catelyn-Stark', 'Samwell-Tarly', 'Tyrion-Lannister', 'Qotho', 'Sandor-Clegane', 'Mordane', 'Lysa-Arryn', 'Jon-Umber-(Greatjon)', 'Vayon-Poole', 'Jorah-Mormont', 'Mycah', 'Jory-Cassel', 'Jeyne-Poole', 'Quaro', 'Stannis-Baratheon', 'Jaime-Lannister', 'Harwin', 'Robert-Baratheon', 'Janos-Slynt']

6-Core: ['Yoren', 'Rickon-Stark', 'Joffrey-Baratheon', 'Janos-Slynt', 'Theon-Greyjoy', 'Varys', 'Sandor-Clegane', 'Bran-Stark', 'Mordane', 'Petyr-Baelish', 'Lysa-Arryn', 'Renly-Baratheon', 'Luwin', 'Sansa-Stark', 'Benjen-Stark', 'Myrcella-Baratheon', 'Edmure-Tully', 'Jeor-Mormont', 'Brynden-Tully', 'Jon-Snow', 'Barristan-Selmy', 'Rodrik-Cassel', 'Tommen-Baratheon', 'Eddard-Stark', 'Gregor-Clegane', 'Meryn-Trant', 'Arya-Stark', 'Loras-Tyrell', 'Jeyne-Poole', 'Jory-Cassel', 'Ilyn-Payne', 'Stannis-Baratheon', 'Cersei-Lannister', 'Jaime-Lannister', 'Jon-Arryn', 'Robb-Stark', 'Rickard-Karstark', 'Tywin-Lannister', 'Pycelle', 'Catelyn-Stark', 'Robert-Baratheon', 'Tyrion-Lannister']

7-Core: ['Rickon-Stark', 'Joffrey-Baratheon', 'Janos-Slynt', 'Theon-Greyjoy', 'Varys', 'Sandor-Clegane', 'Bran-Stark', 'Mordane', 'Petyr-Baelish', 'Lysa-Arryn', 'Renly-Baratheon', 'Luwin', 'Sansa-Stark', 'Benjen-Stark', 'Myrcella-Baratheon', 'Edmure-Tully', 'Brynden-Tully', 'Jon-Snow', 'Barristan-Selmy', 'Rodrik-Cassel', 'Tommen-Baratheon', 'Eddard-Stark', 'Gregor-Clegane', 'Meryn-Trant', 'Arya-Stark', 'Loras-Tyrell', 'Jory-Cassel', 'Ilyn-Payne', 'Stannis-Baratheon', 'Cersei-Lannister', 'Jaime-Lannister', 'Jon-Arryn', 'Robb-Stark', 'Tywin-Lannister', 'Pycelle', 'Catelyn-Stark', 'Robert-Baratheon', 'Tyrion-Lannister']

8-Core: ['Rickon-Stark', 'Joffrey-Baratheon', 'Theon-Greyjoy', 'Varys', 'Sandor-Clegane', 'Bran-Stark', 'Mordane', 'Petyr-Baelish', 'Lysa-Arryn', 'Renly-Baratheon', 'Luwin', 'Sansa-Stark', 'Benjen-Stark', 'Jon-Snow', 'Barristan-Selmy', 'Rodrik-Cassel', 'Eddard-Stark', 'Gregor-Clegane', 'Arya-Stark', 'Loras-Tyrell', 'Jory-Cassel', 'Ilyn-Payne', 'Stannis-Baratheon', 'Cersei-Lannister', 'Jaime-Lannister', 'Jon-Arryn', 'Robb-Stark', 'Tywin-Lannister', 'Pycelle', 'Catelyn-Stark', 'Robert-Baratheon', 'Tyrion-Lannister']

9-Core: ['Rickon-Stark', 'Joffrey-Baratheon', 'Theon-Greyjoy', 'Varys', 'Sandor-Clegane', 'Bran-Stark', 'Mordane', 'Petyr-Baelish', 'Renly-Baratheon', 'Luwin', 'Sansa-Stark', 'Benjen-Stark', 'Jon-Snow', 'Barristan-Selmy', 'Rodrik-Cassel', 'Eddard-Stark', 'Gregor-Clegane', 'Arya-Stark', 'Loras-Tyrell', 'Jory-Cassel', 'Stannis-Baratheon', 'Cersei-Lannister', 'Jaime-Lannister', 'Jon-Arryn', 'Robb-Stark', 'Tywin-Lannister', 'Pycelle', 'Catelyn-Stark', 'Robert-Baratheon', 'Tyrion-Lannister']

10-Core: ['Joffrey-Baratheon', 'Varys', 'Sandor-Clegane', 'Bran-Stark', 'Petyr-Baelish', 'Renly-Baratheon', 'Sansa-Stark', 'Jon-Snow', 'Barristan-Selmy', 'Rodrik-Cassel', 'Eddard-Stark', 'Gregor-Clegane', 'Arya-Stark', 'Loras-Tyrell', 'Jory-Cassel', 'Stannis-Baratheon', 'Cersei-Lannister', 'Jaime-Lannister', 'Jon-Arryn', 'Robb-Stark', 'Tywin-Lannister', 'Pycelle', 'Catelyn-Stark', 'Robert-Baratheon', 'Tyrion-Lannister']

11-Core: ['Joffrey-Baratheon', 'Varys', 'Sandor-Clegane', 'Bran-Stark', 'Petyr-Baelish', 'Renly-Baratheon', 'Sansa-Stark', 'Jon-Snow', 'Barristan-Selmy', 'Rodrik-Cassel', 'Eddard-Stark', 'Arya-Stark', 'Jory-Cassel', 'Stannis-Baratheon', 'Cersei-Lannister', 'Jaime-Lannister', 'Jon-Arryn', 'Robb-Stark', 'Tywin-Lannister', 'Pycelle', 'Catelyn-Stark', 'Robert-Baratheon', 'Tyrion-Lannister']

12-Core: []

13-Core: []

14-Core: []

All found K-Cores in Book 2

2-Core: ['Aegon-I-Targaryen', 'Harren-Hoare', 'Nan', 'Rhaenys-Targaryen', 'Tyrion-Lannister', 'Visenya-Targaryen', 'Aemon-Targaryen-(Maester-Aemon)', 'Jeor-Mormont', 'Jon-Snow', 'Samwell-Tarly', 'Arya-Stark', 'Aeron-Greyjoy', 'Dagmer', 'Theon-Greyjoy', 'Robert-Baratheon', 'Aerys-II-Targaryen', 'Eddard-Stark', 'Jaime-Lannister', 'Rhaegar-Targaryen', 'Aggar', 'Gynir', 'Ramsay-Snow', 'Aggo', 'Belwas', 'Daenerys-Targaryen', 'Jhogo', 'Jorah-Mormont', 'Rakharo', 'Alebelly', 'Bran-Stark', 'Hayhead', 'Jojen-Reed', 'Mikken', 'Alester-Florent', 'Cortnay-Penrose', 'Davos-Seaworth', 'Renly-Baratheon', 'Stannis-Baratheon', 'Allar-Deem', 'Janos-Slynt', 'Allard-Seaworth', 'Alliser-Thorne', 'Joffrey-Baratheon', 'Amabel', 'Harra', 'Amory-Lorch', 'Cersei-Lannister', 'Chiswyck', 'Dunsen', 'Gendry', 'Gregor-Clegane', 'Hot-Pie', 'Ilyn-Payne', 'Meryn-Trant', 'Polliver', 'Rafford', 'Roose-Bolton', 'Sandor-Clegane', 'Tickler', 'Tywin-Lannister', 'Vargo-Hoat', 'Yoren', 'Monford-Velaryon', 'Randyll-Tarly', 'Biter', 'Jaqen-Hghar', 'Koss', 'Lommy-Greenhands', 'Rickon-Stark', 'Robb-Stark', 'Rorge', 'Sansa-Stark', 'Syrio-Forel', 'Weasel', 'Weese', 'Woth', 'Arlys-Oakheart', 'Myrcella-Baratheon', 'Asha-Greyjoy', 'Balon-Greyjoy', 'Wex-Pyke', 'Axell-Florent', 'Selyse-Florent', 'Ygritte', 'Ballabar', 'Podrick-Payne', 'Catelyn-Stark', 'Victarion-Greyjoy', 'Balon-Swann', 'Mandon-Moore', 'Osmund-Kettleblack', 'Barristan-Selmy', 'Varys', 'Benfred-Tallhart', 'Leobald-Tallhart', 'Stygg', 'Beth-Cassel', 'Rodrik-Cassel', 'Boros-Blount', 'Gyles-Rosby', 'Tanda-Stokeworth', 'Tommen-Baratheon', 'Chayle', 'Cley-Cerwyn', 'Donella-Hornwood', 'Hodor', 'Luwin', 'Meera-Reed', 'Osha', 'Reek', 'Stafford-Lannister', 'Walder-Frey-(son-of-Jammos)', 'Walder-Frey-(son-of-Merrett)', 'Wyman-Manderly', 'Brienne-of-Tarth', 'Edmure-Tully', 'Emmon-Cuy', 'Robar-Royce', 'Vyman', 'Bronn', 'Jacelyn-Bywater', 'Timett', 'Bryce-Caron', 'Guyard-Morrigen', 'Brynden-Tully', 'Cleos-Frey', 'Colen-of-Greenpools', 'Desmond-Grell', 'Hallis-Mollen', 'Hoster-Tully', 'Loras-Tyrell', 'Lysa-Arryn', 'Margaery-Tyrell', 'Mathis-Rowan', 'Utherydes-Wayn', 'Wendel-Manderly', 'Lancel-Lannister', 'Osney-Kettleblack', 'Petyr-Baelish', 'Preston-Greenfield', 'Pycelle', 'Shae', 'Vylarr', 'Chella', 'Craster', 'Dywen', 'Eddison-Tollett', 'Mance-Rayder', 'Thoren-Smallwood', 'Cressen', 'Melisandre', 'Patchface', 'Pylos', 'Shireen-Baratheon', 'Drogo', 'Irri', 'Jhiqui', 'Pono', 'Pyat-Pree', 'Viserys-Targaryen', 'Xaro-Xhoan-Daxos', 'Dalbridge', 'Ebben', 'Qhorin-Halfhand', 'Devan-Seaworth', 'Salladhor-Saan', 'Shagga', 'Dontos-Hollard', 'Doran-Martell', 'Drennan', 'Squint', 'Grenn', 'Stonesnake', 'Garlan-Tyrell', 'Lollys-Stokeworth', 'Hallyne', 'Tysha', 'Jaremy-Rykker', 'Jarman-Buckwell', 'Ottyn-Wythers', 'Mace-Tyrell', 'Rattleshirt', 'Lorren', 'Urzen', 'Palla', 'Perwyn-Frey', 'Steffon-Baratheon', 'Walder-Frey', 'Robett-Glover', 'Werlag']
3-Core: ['Aegon-I-Targaryen', 'Harren-Hoare', 'Nan', 'Tyrion-Lannister', 'Aemon-Targaryen-(Maester-Aemon)', 'Jeor-Mormont', 'Jon-Snow', 'Samwell-Tarly', 'Arya-Stark', 'Dagmer', 'Theon-Greyjoy', 'Robert-Baratheon', 'Aerys-II-Targaryen', 'Eddard-Stark', 'Jaime-Lannister', 'Rhaegar-Targaryen', 'Ramsay-Snow', 'Aggo', 'Belwas', 'Daenerys-Targaryen', 'Jhogo', 'Jorah-Mormont', 'Rakharo', 'Alebelly', 'Bran-Stark', 'Jojen-Reed', 'Mikken', 'Alester-Florent', 'Cortnay-Penrose', 'Davos-Seaworth', 'Renly-Baratheon', 'Stannis-Baratheon', 'Janos-Slynt', 'Alliser-Thorne', 'Joffrey-Baratheon', 'Amory-Lorch', 'Cersei-Lannister', 'Chiswyck', 'Dunsen', 'Gendry', 'Gregor-Clegane', 'Hot-Pie', 'Ilyn-Payne']

'Meryn-Trant', 'Polliver', 'Rafford', 'Roose-Bolton', 'Sandor-Clegane',
'Tickler', 'Tywin-Lannister', 'Vargo-Hoat', 'Yoren', 'Randyll-Tarly', 'Biter',
'Jaqen-Hghar', 'Lommy-Greenhands', 'Rickon-Stark', 'Robb-Stark', 'Rorge',
'Sansa-Stark', 'Weasel', 'Weese', 'Woth', 'Arys-Oakheart', 'Myrcella-Baratheon',
'Asha-Greyjoy', 'Balon-Greyjoy', 'Wex-Pyke', 'Selyse-Florent', 'Ygritte',
'Podrick-Payne', 'Catelyn-Stark', 'Balon-Swann', 'Mandon-Moore', 'Osmund-
Kettleblack', 'Barristan-Selmy', 'Varys', 'Beth-Cassel', 'Rodrik-Cassel',
'Boros-Blount', 'Gyles-Rosby', 'Tanda-Stokeworth', 'Tommen-Baratheon', 'Cley-
Cerwyn', 'Donella-Hornwood', 'Hodor', 'Luwin', 'Meera-Reed', 'Osha', 'Stafford-
Lannister', 'Walder-Frey-(son-of-Merrett)', 'Wyman-Manderly', 'Brienne-of-
Tarth', 'Edmure-Tully', 'Emmon-Cuy', 'Robar-Royce', 'Vyman', 'Bronn', 'Jacelyn-
Bywater', 'Timett', 'Bryce-Caron', 'Brynden-Tully', 'Cleos-Frey', 'Hoster-
Tully', 'Loras-Tyrell', 'Lysa-Arryn', 'Margaery-Tyrell', 'Mathis-Rowan',
'Lancel-Lannister', 'Petyr-Baelish', 'Pycelle', 'Shae', 'Chella', 'Craster',
'Dywen', 'Eddison-Tollett', 'Mance-Rayder', 'Thoren-Smallwood', 'Cressen',
'Melisandre', 'Pylos', 'Drogo', 'Pono', 'Viserys-Targaryen', 'Dalbridge',
'Ebbin', 'Qhorin-Halfhand', 'Devan-Seaworth', 'Shagga', 'Dontos-Hollard',
'Doran-Martell', 'Grenn', 'Stonesnake', 'Tysha', 'Jarman-Buckwell', 'Ottyn-
Wythers', 'Mace-Tyrell', 'Rattleshirt']

4-Core: ['Amory-Lorch', 'Rickon-Stark', 'Mace-Tyrell', 'Rorge', 'Belwas',
'Luwin', 'Gendry', 'Mathis-Rowan', 'Jeor-Mormont', 'Tanda-Stokeworth',
'Barristan-Selmy', 'Gyles-Rosby', 'Dunsen', 'Dontos-Hollard', 'Meera-Reed',
'Tommen-Baratheon', 'Eddard-Stark', 'Meryn-Trant', 'Mance-Rayder', 'Brienne-of-
Tarth', 'Cressen', 'Osmund-Kettleblack', 'Cersei-Lannister', 'Walder-Frey-(son-
of-Merrett)', 'Jojen-Reed', 'Margaery-Tyrell', 'Robb-Stark', 'Davos-Seaworth',
'Joffrey-Baratheon', 'Bronn', 'Varys', 'Lancel-Lannister', 'Edmure-Tully',
'Jacelyn-Bywater', 'Jon-Snow', 'Thoren-Smallwood', 'Randyll-Tarly', 'Roose-
Bolton', 'Osha', 'Weese', 'Donella-Hornwood', 'Daenerys-Targaryen', 'Alliser-
Thorne', 'Jhogo', 'Arya-Stark', 'Rafford', 'Woth', 'Boros-Blount', 'Loras-
Tyrell', 'Illyn-Payne', 'Balon-Swann', 'Tywin-Lannister', 'Pycelle', 'Weasel',
'Tickler', 'Yoren', 'Cleos-Frey', 'Stafford-Lannister', 'Theon-Greyjoy',
'Craster', 'Bran-Stark', 'Petyr-Baelish', 'Renly-Baratheon', 'Shae', 'Biter',
'Sansa-Stark', 'Myrcella-Baratheon', 'Aerys-II-Targaryen', 'Rodrik-Cassel',
'Polliver', 'Gregor-Clegane', 'Aggo', 'Hodor', 'Catelyn-Stark', 'Melisandre',
'Tyrion-Lannister', 'Jaqen-Hghar', 'Chiswyck', 'Qhorin-Halfhand', 'Sandor-
Clegane', 'Balon-Greyjoy', 'Jorah-Mormont', 'Hot-Pie', 'Ramsay-Snow', 'Cortnay-
Penrose', 'Lommy-Greenhands', 'Alester-Florent', 'Stannis-Baratheon', 'Mandon-
Moore', 'Jaime-Lannister', 'Wyman-Manderly', 'Robert-Baratheon', 'Vargo-Hoat',
'Janos-Slynt']

5-Core: ['Amory-Lorch', 'Yoren', 'Cleos-Frey', 'Rickon-Stark', 'Chiswyck',
'Joffrey-Baratheon', 'Bronn', 'Theon-Greyjoy', 'Varys', 'Sandor-Clegane', 'Bran-
Stark', 'Lancel-Lannister', 'Balon-Greyjoy', 'Petyr-Baelish', 'Renly-Baratheon',
'Luwin', 'Gendry', 'Sansa-Stark', 'Myrcella-Baratheon', 'Edmure-Tully',
'Jacelyn-Bywater', 'Jon-Snow', 'Tanda-Stokeworth', 'Gyles-Rosby', 'Rodrik-
Cassel', 'Dunsen', 'Polliver', 'Roose-Bolton', 'Osha', 'Meera-Reed', 'Tommen-
Baratheon', 'Weese', 'Eddard-Stark', 'Gregor-Clegane', 'Meryn-Trant', 'Rafford',
'Arya-Stark', 'Woth', 'Hot-Pie', 'Boros-Blount', 'Ramsay-Snow', 'Brienne-of-
Tarth', 'Tyrion-Lannister', 'Lommy-Greenhands', 'Loras-Tyrell', 'Osmund-

Kettleblack', 'Ilyn-Payne', 'Hodor', 'Mandon-Moore', 'Stannis-Baratheon',
 'Cersei-Lannister', 'Jaime-Lannister', 'Jojen-Reed', 'Margaery-Tyrell', 'Robb-Stark',
 'Balon-Swann', 'Tywin-Lannister', 'Pycelle', 'Catelyn-Stark', 'Robert-Baratheon',
 'Janos-Slynt', 'Tickler']
 6-Core: ['Amory-Lorch', 'Cleos-Frey', 'Rickon-Stark', 'Chiswyck', 'Joffrey-Baratheon',
 'Bronn', 'Theon-Greyjoy', 'Varys', 'Sandor-Clegane', 'Bran-Stark',
 'Lancel-Lannister', 'Petyr-Baelish', 'Renly-Baratheon', 'Luwin', 'Sansa-Stark',
 'Myrcella-Baratheon', 'Jacelyn-Bywater', 'Jon-Snow', 'Gyles-Rosby', 'Rodrik-Cassel',
 'Dunsen', 'Polliver', 'Osha', 'Meera-Reed', 'Tommen-Baratheon',
 'Weese', 'Eddard-Stark', 'Gregor-Clegane', 'Meryn-Trant', 'Rafford', 'Arya-Stark',
 'Boros-Blount', 'Tyrion-Lannister', 'Loras-Tyrell', 'Ilyn-Payne',
 'Hodor', 'Mandon-Moore', 'Stannis-Baratheon', 'Cersei-Lannister', 'Jaime-Lannister',
 'Jojen-Reed', 'Robb-Stark', 'Tywin-Lannister', 'Pycelle', 'Catelyn-Stark', 'Robert-Baratheon',
 'Janos-Slynt', 'Tickler']
 7-Core: ['Amory-Lorch', 'Rickon-Stark', 'Chiswyck', 'Joffrey-Baratheon',
 'Bronn', 'Varys', 'Sandor-Clegane', 'Bran-Stark', 'Lancel-Lannister', 'Petyr-Baelish',
 'Renly-Baratheon', 'Sansa-Stark', 'Myrcella-Baratheon', 'Jacelyn-Bywater',
 'Jon-Snow', 'Gyles-Rosby', 'Polliver', 'Dunsen', 'Tommen-Baratheon',
 'Eddard-Stark', 'Gregor-Clegane', 'Meryn-Trant', 'Rafford', 'Arya-Stark',
 'Boros-Blount', 'Tyrion-Lannister', 'Ilyn-Payne', 'Stannis-Baratheon', 'Cersei-Lannister',
 'Jaime-Lannister', 'Robb-Stark', 'Tywin-Lannister', 'Pycelle', 'Catelyn-Stark', 'Robert-Baratheon',
 'Janos-Slynt', 'Tickler']
 8-Core: ['Amory-Lorch', 'Chiswyck', 'Joffrey-Baratheon', 'Varys', 'Sandor-Clegane',
 'Petyr-Baelish', 'Renly-Baratheon', 'Sansa-Stark', 'Polliver', 'Dunsen', 'Eddard-Stark',
 'Gregor-Clegane', 'Meryn-Trant', 'Rafford', 'Arya-Stark', 'Ilyn-Payne', 'Stannis-Baratheon',
 'Cersei-Lannister', 'Jaime-Lannister', 'Robb-Stark', 'Tywin-Lannister', 'Catelyn-Stark', 'Robert-Baratheon',
 'Tyrion-Lannister', 'Tickler']
 9-Core: ['Amory-Lorch', 'Chiswyck', 'Joffrey-Baratheon', 'Varys', 'Sandor-Clegane',
 'Petyr-Baelish', 'Renly-Baratheon', 'Sansa-Stark', 'Polliver', 'Dunsen', 'Eddard-Stark',
 'Gregor-Clegane', 'Meryn-Trant', 'Rafford', 'Arya-Stark', 'Ilyn-Payne', 'Stannis-Baratheon',
 'Cersei-Lannister', 'Jaime-Lannister', 'Robb-Stark', 'Tywin-Lannister', 'Catelyn-Stark', 'Robert-Baratheon',
 'Tyrion-Lannister', 'Tickler']
 10-Core: ['Amory-Lorch', 'Chiswyck', 'Joffrey-Baratheon', 'Sandor-Clegane',
 'Renly-Baratheon', 'Sansa-Stark', 'Polliver', 'Dunsen', 'Eddard-Stark', 'Gregor-Clegane',
 'Meryn-Trant', 'Rafford', 'Arya-Stark', 'Ilyn-Payne', 'Stannis-Baratheon', 'Cersei-Lannister',
 'Jaime-Lannister', 'Robb-Stark', 'Tywin-Lannister', 'Catelyn-Stark', 'Robert-Baratheon',
 'Tyrion-Lannister', 'Tickler']
 11-Core: ['Amory-Lorch', 'Tickler', 'Cersei-Lannister', 'Polliver', 'Chiswyck',
 'Joffrey-Baratheon', 'Gregor-Clegane', 'Meryn-Trant', 'Rafford', 'Sandor-Clegane',
 'Ilyn-Payne', 'Dunsen']
 12-Core: []
 13-Core: []
 14-Core: []

All found K-Cores in Book 3

2-Core: ['Addam-Marbrand', 'Gyles-Rosby', 'Jaime-Lannister', 'Jalabhar-Xho',

'Joffrey-Baratheon', 'Kevan-Lannister', 'Oberyn-Martell', 'Tyrion-Lannister',
'Varys', 'Aegon-Frey-(son-of-Stevron)', 'Catelyn-Stark', 'Robb-Stark', 'Stevron-Frey',
'Walder-Frey', 'Aegon-I-Targaryen', 'Daenerys-Targaryen', 'Stannis-Baratheon',
'Torren-Stark', 'Elia-Martell', 'Aemon-Targaryen-(Maester-Aemon)',
'Alliser-Thorne', 'Clydas', 'Cotter-Pyke', 'Denys-Mallister', 'Donal-Noye',
'Eddison-Tollett', 'Grenn', 'Hobb', 'Janos-Slynt', 'Jeor-Mormont', 'Jon-Snow',
'Mance-Rayder', 'Owen', 'Pypar', 'Robert-Baratheon', 'Samwell-Tarly', 'Aerys-II-Targaryen',
'Arthur-Dayne', 'Brienne-of-Tarth', 'Tywin-Lannister', 'Viserys-Targaryen',
'Aggo', 'Jhogo', 'Rakharo', 'Alayaya', 'Bronn', 'Cersei-Lannister',
'Alerie-Hightower', 'Mace-Tyrell', 'Margaery-Tyrell', 'Alester-Florent', 'Axell-Florent',
'Davos-Seaworth', 'Lamprey', 'Selyse-Florent', 'Alla-Tyrell', 'Elinor-Tyrell',
'Megga-Tyrell', 'Sansa-Stark', 'Qhorin-Halfhand', 'Amory-Lorch',
'Gregor-Clegane', 'Vargo-Hoat', 'Andrew-Estermont', 'Edric-Storm', 'Anguy',
'Arya-Stark', 'Beric-Dondarrion', 'Greenbeard', 'Harwin', 'Husband', 'Lem',
'Sandor-Clegane', 'Sharna', 'Tom-of-Sevenstreams', 'Arstan', 'Belwas', 'Illyrio-Mopatis',
'Jhiqui', 'Jorah-Mormont', 'Kraznys-mo-Nakloz', 'Mero', 'Rhaegar-Targaryen',
'Gerold-Hightower', 'Lewyn-Martell', 'Bran-Stark', 'Brynden-Tully',
'Dunsen', 'Eddard-Stark', 'Elmar-Frey', 'Gendry', 'Hot-Pie', 'Ilyn-Payne',
'Meryn-Trant', 'Mycah', 'Nan', 'Polliver', 'Rafford', 'Rickon-Stark', 'Roose-Bolton',
'Thoros-of-Myr', 'Tickler', 'Melisandre', 'Porridge', 'Salladhor-Saan',
'Baelor-I-Targaryen', 'Balon-Greyjoy', 'Balon-Swann', 'Boros-Blount', 'Loras-Tyrell',
'Osmund-Kettleblack', 'Bannen', 'Dywen', 'Barristan-Selmy', 'Bedwyck',
'Craster', 'Ben-Plumm', 'Oznak-zo-Pahl', 'Daario-Naharis', 'Benjen-Stark',
'Jack-Bulwer', 'Big-Boil', 'Ygritte', 'Rorge', 'Tommen-Baratheon', 'Bowen-Marsh',
'Othell-Yarwyck', 'Coldhands', 'Hodor', 'Jojen-Reed', 'Jon-Umber-(Greatjon)',
'Meera-Reed', 'Theon-Greyjoy', 'Brandon-Stark', 'Brella',
'Shae', 'Cleos-Frey', 'Qyburn', 'Walton', 'Lolllys-Stokeworth', 'Mandon-Moore',
'Podrick-Payne', 'Edmure-Tully', 'Rickard-Karstark', 'Butterbumps', 'Olenna-Redwyne',
'Dacey-Mormont', 'Desmond-Grell', 'Hoster-Tully', 'Jeyne-Westerling',
'Lothar-Frey', 'Lysa-Arryn', 'Maege-Mormont', 'Marq-Piper', 'Petyr-Baelish',
'Raynald-Westerling', 'Robin-Ryger', 'Roslin-Frey', 'Ryman-Frey', 'Tansy',
'Vyman', 'Ellaria-Sand', 'Lancel-Lannister', 'Pycelle', 'Willas-Tyrell',
'Chett', 'Lark', 'Small-Paul', 'Cleon', 'Ghelo', 'Clubfoot-Karl', 'Gilly',
'Dirk', 'Tormund', 'Drogo', 'Irri', 'Prendahl-na-Ghezn', 'Sallor', 'Grey-Worm',
'Missandei', 'Shireen-Baratheon', 'Dalla', 'Val', 'Devan-Seaworth', 'Guncer-Sunglass',
'Khorane-Sathmantes', 'Pylos', 'Utherydes-Wayn', 'Dick-Follard',
'Satin', 'Mag-Mar-Tun-Doh-Weg', 'Dontos-Hollard', 'Lothor-Brune', 'Doran-Martell',
'Errok', 'Grigg', 'Styr', 'Robett-Glover', 'Garlan-Tyrell', 'Gawen-Westerling',
'Sybell-Spicer', 'Gendel', 'Gorne', 'Helman-Tallhart', 'Kegs',
'Jarl', 'Hareth-(Moles-Town)', 'Mully', 'Spare-Boot', 'Harma', 'Rattleshirt',
'Varamyr', 'Weeper', 'Randyll-Tarly', 'High-Septon-(Tyrions)', 'Renly-Baratheon',
'Shagwell', 'Thoren-Smallwood', 'Myrcella-Baratheon', 'Paxter-Redwyne',
'Jon-Arryn', 'Robert-Arryn', 'Orell', 'Ryk', 'Jon-Umber-(Smalljon)',
'Robin-Flint', 'Wendel-Manderly', 'Jonothor-Darry', 'Oswell-Whent', 'Willem-Lannister',
'Symon-Silver-Tongue', 'Tanda-Stokeworth', 'Merrett-Frey', 'Walder-Rivers',
'Marillion', 'Mathis-Rowan', 'Martyn-Lannister', 'Olyvar-Frey',
'Oswell-Kettleblack', 'Petyr-Frey', 'Ramsay-Snow', 'Rollam-Westerling',
'Rhaella-Targaryen', 'Zollo', 'Tysha']

3-Core: ['Addam-Marbrand', 'Jaime-Lannister', 'Joffrey-Baratheon', 'Kevan-Lannister', 'Oberyn-Martell', 'Tyrion-Lannister', 'Varys', 'Aegon-Frey-(son-of-Stevron)', 'Catelyn-Stark', 'Robb-Stark', 'Stevron-Frey', 'Walder-Frey', 'Daenerys-Targaryen', 'Stannis-Baratheon', 'Elia-Martell', 'Aemon-Targaryen-(Maester-Aemon)', 'Alliser-Thorne', 'Clydas', 'Cotter-Pyke', 'Denys-Mallister', 'Donal-Noye', 'Eddison-Tollett', 'Grenn', 'Hobb', 'Janos-Slynt', 'Jeor-Mormont', 'Jon-Snow', 'Mance-Rayder', 'Owen', 'Pypar', 'Robert-Baratheon', 'Samwell-Tarly', 'Aerys-II-Targaryen', 'Arthur-Dayne', 'Brienne-of-Tarth', 'Tywin-Lannister', 'Viserys-Targaryen', 'Alayaya', 'Bronn', 'Cersei-Lannister', 'Mace-Tyrell', 'Margaery-Tyrell', 'Alester-Florent', 'Axell-Florent', 'Davos-Seaworth', 'Lamprey', 'Selyse-Florent', 'Alla-Tyrell', 'Elinor-Tyrell', 'Megga-Tyrell', 'Sansa-Stark', 'Qhorin-Halfhand', 'Amory-Lorch', 'Gregor-Clegane', 'Vargo-Hoat', 'Edric-Storm', 'Anguy', 'Arya-Stark', 'Beric-Dondarrion', 'Greenbeard', 'Harwin', 'Lem', 'Sandor-Clegane', 'Sharna', 'Tom-of-Sevenstreams', 'Arstan', 'Belwas', 'Jhiqui', 'Jorah-Mormont', 'Mero', 'Rhaegar-Targaryen', 'Gerold-Hightower', 'Bran-Stark', 'Brynden-Tully', 'Dunsen', 'Eddard-Stark', 'Elmar-Frey', 'Gendry', 'Hot-Pie', 'Ilyn-Payne', 'Meryn-Trant', 'Mycah', 'Polliver', 'Rafford', 'Rickon-Stark', 'Roose-Bolton', 'Thoros-of-Myr', 'Tickler', 'Melisandre', 'Porridge', 'Salladhor-Saan', 'Baelor-I-Targaryen', 'Balon-Greyjoy', 'Balon-Swann', 'Boros-Blount', 'Loras-Tyrell', 'Osmund-Kettleblack', 'Dywen', 'Barristan-Selmy', 'Bedwyck', 'Craster', 'Ben-Plumm', 'Daario-Naharis', 'Benjen-Stark', 'Ygritte', 'Tommen-Baratheon', 'Bowen-Marsh', 'Othell-Yarwyck', 'Hodor', 'Jojen-Reed', 'Jon-Umber-(Greatjon)', 'Meera-Reed', 'Theon-Greyjoy', 'Brandon-Stark', 'Shae', 'Cleos-Frey', 'Qyburn', 'Walton', 'Lollys-Stokeworth', 'Mandon-Moore', 'Podrick-Payne', 'Edmure-Tully', 'Rickard-Karstark', 'Olenna-Redwyne', 'Dacey-Mormont', 'Desmond-Grell', 'Hoster-Tully', 'Jeyne-Westerling', 'Lothar-Frey', 'Lysa-Arryn', 'Maege-Mormont', 'Marq-Piper', 'Petyr-Baelish', 'Roslin-Frey', 'Ryman-Frey', 'Tansy', 'Ellaria-Sand', 'Pycelle', 'Willas-Tyrell', 'Chett', 'Lark', 'Small-Paul', 'Gilly', 'Tormund', 'Drogo', 'Irri', 'Missandei', 'Shireen-Baratheon', 'Dalla', 'Val', 'Devan-Seaworth', 'Khorane-Sathmantes', 'Pylos', 'Satin', 'Dontos-Hollard', 'Lothor-Brone', 'Doran-Martell', 'Errok', 'Grigg', 'Styr', 'Garlan-Tyrell', 'Kegs', 'Jarl', 'Hareth-(Moles-Town)', 'Mully', 'Spare-Boot', 'Harma', 'Rattleshirt', 'Varamyr', 'High-Septon-(Tyrions)', 'Renly-Baratheon', 'Myrcella-Baratheon', 'Paxter-Redwyne', 'Jon-Arryn', 'Robert-Arryn', 'Ryk', 'Symon-Silver-Tongue', 'Tanda-Stokeworth', 'Marillion']

4-Core: ['Amory-Lorch', 'Beric-Dondarrion', 'Marq-Piper', 'Rickon-Stark', 'Willas-Tyrell', 'Lark', 'Mace-Tyrell', 'Mully', 'Greenbeard', 'Drogo', 'Belwas', 'Gendry', 'Walder-Frey', 'Jeor-Mormont', 'Irri', 'Olenna-Redwyne', 'Tanda-Stokeworth', 'Barristan-Selmy', 'Dunsen', 'Dontos-Hollard', 'Rhaegar-Targaryen', 'Meera-Reed', 'Tommen-Baratheon', 'Eddard-Stark', 'Meryn-Trant', 'Mance-Rayder', 'Brienne-of-Tarth', 'Ryk', 'Osmund-Kettleblack', 'Small-Paul', 'Kevan-Lannister', 'Cersei-Lannister', 'Missandei', 'Margaery-Tyrell', 'Jojen-Reed', 'Robb-Stark', 'Rickard-Karstark', 'Qyburn', 'Davos-Seaworth', 'Bowen-Marsh', 'Clydas', 'Eddison-Tollett', 'Viserys-Targaryen', 'Joffrey-Baratheon', 'Bronn', 'Ygritte', 'Varys', 'Jhiqui', 'Anguy', 'Jarl', 'Edmure-Tully', 'Shireen-Baratheon', 'Aemon-Targaryen-(Maester-Aemon)', 'Brynden-Tully', 'Pypar', 'Tom-of-Sevenstreams', 'Jon-Snow', 'Hobb', 'Owen', 'Podrick-Payne']

'Roose-Bolton', 'Elia-Martell', 'Daenerys-Targaryen', 'Alliser-Thorne',
'Rafford', 'Arya-Stark', 'Boros-Blount', 'Loras-Tyrell', 'Ilyn-Payne', 'Balon-Swann', 'Tywin-Lannister', 'Pycelle', 'Donal-Noye', 'Edric-Storm', 'Tickler', 'Theon-Greyjoy', 'Pylos', 'Gilly', 'Craster', 'Bran-Stark', 'Petyr-Baelish', 'Renly-Baratheon', 'Shae', 'Sansa-Stark', 'Selyse-Florent', 'Benjen-Stark', 'Myrcella-Baratheon', 'Grenn', 'Spare-Boot', 'Dywen', 'Elmar-Frey', 'Salladhor-Saan', 'Hareth-(Moles-Town)', 'Axell-Florent', 'Aerys-II-Targaryen', 'Walton', 'Kegs', 'Polliver', 'Jeyne-Westerling', 'Daario-Naharis', 'Gregor-Clegane', 'Tormund', 'Lollys-Stokeworth', 'Oberyn-Martell', 'Lothar-Frey', 'Arstan', 'Hoster-Tully', 'Garlan-Tyrell', 'Addam-Marbrand', 'Catelyn-Stark', 'Samwell-Tarly', 'Rattleshirt', 'Lem', 'Melisandre', 'Tyrion-Lannister', 'Chett', 'Qhorin-Halfhand', 'Denys-Mallister', 'Sandor-Clegane', 'Bedwyck', 'Satin', 'Lysa-Arryn', 'Roslin-Frey', 'Jon-Umber-(Greatjon)', 'Cotter-Pyke', 'Thoros-of-Myr', 'Othell-Yarwyck', 'Jorah-Mormont', 'Styr', 'Hot-Pie', 'Tansy', 'Alester-Florent', 'Stannis-Baratheon', 'Mandon-Moore', 'Sharna', 'Jaime-Lannister', 'Harma', 'Harwin', 'Robert-Baratheon', 'Vargo-Hoat', 'Janos-Slynt']

5-Core: ['Beric-Dondarrion', 'Willas-Tyrell', 'Rickon-Stark', 'Mace-Tyrell', 'Mully', 'Greenbeard', 'Gendry', 'Walder-Frey', 'Jeor-Mormont', 'Olenna-Redwyne', 'Dunsen', 'Dontos-Hollard', 'Tommen-Baratheon', 'Eddard-Stark', 'Meryn-Trant', 'Mance-Rayder', 'Brienne-of-Tarth', 'Osmund-Kettleblack', 'Kevan-Lannister', 'Cersei-Lannister', 'Margaery-Tyrell', 'Rickard-Karstark', 'Robb-Stark', 'Bowen-Marsh', 'Eddison-Tollett', 'Joffrey-Baratheon', 'Bronn', 'Ygritte', 'Varys', 'Anguy', 'Jarl', 'Edmure-Tully', 'Aemon-Targaryen-(Maester-Aemon)', 'Brynden-Tully', 'Pypar', 'Tom-of-Sevenstreams', 'Jon-Snow', 'Hobb', 'Owen', 'Roose-Bolton', 'Elia-Martell', 'Alliser-Thorne', 'Rafford', 'Arya-Stark', 'Boros-Blount', 'Loras-Tyrell', 'Ilyn-Payne', 'Balon-Swann', 'Tywin-Lannister', 'Pycelle', 'Donal-Noye', 'Tickler', 'Craster', 'Bran-Stark', 'Petyr-Baelish', 'Renly-Baratheon', 'Shae', 'Sansa-Stark', 'Grenn', 'Spare-Boot', 'Dywen', 'Elmar-Frey', 'Hareth-(Moles-Town)', 'Aerys-II-Targaryen', 'Kegs', 'Polliver', 'Gregor-Clegane', 'Tormund', 'Oberyn-Martell', 'Lothar-Frey', 'Hoster-Tully', 'Garlan-Tyrell', 'Addam-Marbrand', 'Catelyn-Stark', 'Samwell-Tarly', 'Rattleshirt', 'Lem', 'Tyrion-Lannister', 'Qhorin-Halfhand', 'Denys-Mallister', 'Sandor-Clegane', 'Bedwyck', 'Satin', 'Lysa-Arryn', 'Roslin-Frey', 'Jon-Umber-(Greatjon)', 'Cotter-Pyke', 'Thoros-of-Myr', 'Othell-Yarwyck', 'Styr', 'Tansy', 'Stannis-Baratheon', 'Jaime-Lannister', 'Harma', 'Harwin', 'Robert-Baratheon', 'Janos-Slynt']

6-Core: ['Beric-Dondarrion', 'Willas-Tyrell', 'Mace-Tyrell', 'Greenbeard', 'Gendry', 'Walder-Frey', 'Jeor-Mormont', 'Olenna-Redwyne', 'Dunsen', 'Tommen-Baratheon', 'Eddard-Stark', 'Meryn-Trant', 'Mance-Rayder', 'Brienne-of-Tarth', 'Osmund-Kettleblack', 'Kevan-Lannister', 'Cersei-Lannister', 'Margaery-Tyrell', 'Robb-Stark', 'Bowen-Marsh', 'Eddison-Tollett', 'Joffrey-Baratheon', 'Bronn', 'Ygritte', 'Varys', 'Anguy', 'Jarl', 'Edmure-Tully', 'Aemon-Targaryen-(Maester-Aemon)', 'Brynden-Tully', 'Pypar', 'Tom-of-Sevenstreams', 'Jon-Snow', 'Hobb', 'Owen', 'Roose-Bolton', 'Alliser-Thorne', 'Rafford', 'Arya-Stark', 'Boros-Blount', 'Loras-Tyrell', 'Ilyn-Payne', 'Balon-Swann', 'Tywin-Lannister', 'Pycelle', 'Donal-Noye', 'Tickler', 'Craster', 'Bran-Stark', 'Petyr-Baelish', 'Renly-Baratheon', 'Sansa-Stark', 'Grenn', 'Elmar-Frey', 'Aerys-II-Targaryen', 'Polliver', 'Gregor-Clegane', 'Tormund', 'Oberyn-Martell', 'Lothar-Frey',

'Hoster-Tully', 'Addam-Marbrand', 'Catelyn-Stark', 'Samwell-Tarly',
 'Rattleshirt', 'Lem', 'Tyrion-Lannister', 'Qhorin-Halfhand', 'Denys-Mallister',
 'Sandor-Clegane', 'Lysa-Arryn', 'Cotter-Pyke', 'Thoros-of-Myr', 'Othell-
 Yarwyck', 'Styr', 'Tansy', 'Stannis-Baratheon', 'Jaime-Lannister', 'Harma',
 'Harwin', 'Robert-Baratheon', 'Janos-Slynt']
 7-Core: ['Mace-Tyrell', 'Joffrey-Baratheon', 'Bronn', 'Varys', 'Denys-
 Mallister', 'Sandor-Clegane', 'Bran-Stark', 'Petyr-Baelish', 'Lysa-Arryn',
 'Renly-Baratheon', 'Sansa-Stark', 'Jeor-Mormont', 'Aemon-Targaryen-(Maester-
 Aemon)', 'Pypar', 'Olenna-Redwyne', 'Jon-Snow', 'Elmar-Frey', 'Cotter-Pyke',
 'Othell-Yarwyck', 'Dunsen', 'Polliver', 'Roose-Bolton', 'Tommen-Baratheon',
 'Eddard-Stark', 'Gregor-Clegane', 'Meryn-Trant', 'Rafford', 'Arya-Stark',
 'Brienne-of-Tarth', 'Tyrion-Lannister', 'Loras-Tyrell', 'Osmund-Kettleblack',
 'Oberyn-Martell', 'Ilyn-Payne', 'Kevan-Lannister', 'Stannis-Baratheon', 'Cersei-
 Lannister', 'Jaime-Lannister', 'Margaery-Tyrell', 'Robb-Stark', 'Eddison-
 Tollett', 'Tywin-Lannister', 'Catelyn-Stark', 'Robert-Baratheon', 'Bowen-Marsh',
 'Samwell-Tarly', 'Janos-Slynt', 'Tickler']
 8-Core: ['Joffrey-Baratheon', 'Bronn', 'Varys', 'Sandor-Clegane', 'Sansa-Stark',
 'Dunsen', 'Polliver', 'Tommen-Baratheon', 'Eddard-Stark', 'Gregor-Clegane',
 'Meryn-Trant', 'Rafford', 'Arya-Stark', 'Oberyn-Martell', 'Ilyn-Payne',
 'Stannis-Baratheon', 'Cersei-Lannister', 'Jaime-Lannister', 'Robb-Stark',
 'Tywin-Lannister', 'Catelyn-Stark', 'Robert-Baratheon', 'Tyrion-Lannister',
 'Tickler']
 9-Core: ['Tickler', 'Cersei-Lannister', 'Polliver', 'Gregor-Clegane', 'Joffrey-
 Baratheon', 'Meryn-Trant', 'Rafford', 'Arya-Stark', 'Sandor-Clegane', 'Ilyn-
 Payne', 'Dunsen']
 10-Core: ['Tickler', 'Cersei-Lannister', 'Polliver', 'Gregor-Clegane', 'Joffrey-
 Baratheon', 'Meryn-Trant', 'Rafford', 'Arya-Stark', 'Sandor-Clegane', 'Ilyn-
 Payne', 'Dunsen']
 11-Core: []
 12-Core: []
 13-Core: []
 14-Core: []

All found K-Cores in Book 4

2-Core: ['Addam-Marbrand', 'Brynden-Tully', 'Cersei-Lannister', 'Jaime-
 Lannister', 'Lyle-Crakehall', 'Rhaegar-Targaryen', 'Aemon-Targaryen-(Maester-
 Aemon)', 'Alleras', 'Clydas', 'Dareon', 'Gilly', 'Jon-Snow', 'Samwell-Tarly',
 'Aeron-Greyjoy', 'Asha-Greyjoy', 'Baelor-Blacktyde', 'Balon-Greyjoy', 'Dunstan-
 Drumm', 'Euron-Greyjoy', 'Gorold-Goodbrother', 'Meldred-Merlyn', 'Murenmure',
 'Theon-Greyjoy', 'Victarion-Greyjoy', 'Aerys-II-Targaryen', 'Robert-Baratheon',
 'Tywin-Lannister', 'Alla-Tyrell', 'Elinor-Tyrell', 'Margaery-Tyrell', 'Megga-
 Tyrell', 'Osney-Kettleblack', 'Armen', 'Leo-Tyrell', 'Marwyn', 'Mollander',
 'Pate-(novice)', 'Roone', 'Jon-Arryn', 'Amerei-Frey', 'Lancel-Lannister',
 'Quentyn-Martell', 'Andrey-Dalt', 'Arianne-Martell', 'Garin-(orphan)',
 'Myrcella-Baratheon', 'Sylva-Santagar', 'Tyene-Sand', 'Anya-Waynwood', 'Benedar-
 Belmore', 'Gilwood-Hunter', 'Harrold-Hardyng', 'Horton-Redfort', 'Petyr-
 Baelish', 'Symond-Templeton', 'Yohn-Royce', 'Areo-Hotah', 'Arys-Oakheart',
 'Caleotte', 'Doran-Martell', 'Obara-Sand', 'Oberyn-Martell', 'Cedra', 'Gerold-'

Dayne', 'Mellei', 'Morra', 'Arya-Stark', 'Brea', 'Brusco', 'Denyo-Terys', 'Kindly-Man', 'Sandor-Clegane', 'Sansa-Stark', 'Talea', 'Ternesio-Terys', 'Waif', 'Red-Oarsman', 'Rodrik-Harlaw', 'Aurane-Waters', 'Gyles-Rosby', 'Harys-Swyft', 'Loras-Tyrell', 'Orton-Merryweather', 'Paxter-Redwyne', 'Stannis-Baratheon', 'Tommen-Baratheon', 'Balman-Byrch', 'Bronn', 'Falyse-Stokeworth', 'Balon-Swann', 'Gregor-Clegane', 'Beric-Dondarrion', 'Catelyn-Stark', 'Mariya-Darry', 'Thoros-of-Myr', 'Brienne-of-Tarth', 'Blue-Bard', 'Lambert-Turnberry', 'Qyburn', 'Tallad', 'Boros-Blount', 'Meryn-Trant', 'Osmund-Kettleblack', 'Bran-Stark', 'Rickon-Stark', 'Robb-Stark', 'Creighton-Longbough', 'Dick-Crabb', 'Dontos-Hollard', 'Elder-Brother', 'Gendry', 'Hyle-Hunt', 'Illifer', 'Joffrey-Baratheon', 'Lysa-Arryn', 'Meribald', 'Narbert', 'Podrick-Payne', 'Pyg', 'Randyll-Tarly', 'Renly-Baratheon', 'Shagwell', 'Timeon', 'Tyrion-Lannister', 'Willow-Heddle', 'Daven-Lannister', 'Edmure-Tully', 'Edwyn-Frey', 'Emmon-Frey', 'Ryman-Frey', 'Dorcias', 'Dunsen', 'Eddard-Stark', 'High-Septon-(Tyrions)', 'High-Sparrow', 'Ilyn-Payne', 'Jalabhar-Xho', 'Jocelyn-Swyft', 'Kevan-Lannister', 'Mace-Tyrell', 'Maegor-I-Targaryen', 'Maggy', 'Melara-Hetherspoon', 'Moon-Boy', 'Olenna-Redwyne', 'Osflyd-Kettleblack', 'Pycelle', 'Rafford', 'Senelle', 'Taena-of-Myr', 'Varys', 'Walder-Frey', 'Colemon', 'Gretchel', 'Maddy', 'Robert-Arryn', 'Craster', 'Jeor-Mormont', 'Shadrich', 'Daemon-Sand', 'Nymeria-Sand', 'Eddison-Tollett', 'Forley-Prest', 'Roslin-Frey', 'Walder-Rivers', 'Genna-Lannister', 'Hotho-Harlaw', 'Humfrey-Hewett', 'Nute', 'Quellon-Greyjoy', 'Terrance-Lynderly', 'Kojja-Mo', 'Melisandre', 'Grenn', 'Pypar', 'Gyles-Grafton', 'Sigflyd-Harlaw', 'Josmyn-Peckledon', 'Pia', 'Vargo-Hoat', 'Zollo', 'Lothor-Brune', 'Mya-Stone', 'Lyn-Corbray', 'Lyonel-Corbray', 'Marillion', 'Nestor-Royce', 'Myranda-Royce', 'Trystane-Martell', 'Ralf-Stonehouse', 'Quhuru-Mo', 'Xhondo']

3-Core: ['Addam-Marbrand', 'Brynden-Tully', 'Cersei-Lannister', 'Jaime-Lannister', 'Lyle-Crakehall', 'Rhaegar-Targaryen', 'Aemon-Targaryen-(Maester-Aemon)', 'Alleras', 'Dareon', 'Gilly', 'Jon-Snow', 'Samwell-Tarly', 'Aeron-Greyjoy', 'Asha-Greyjoy', 'Baelor-Blacktyde', 'Balon-Greyjoy', 'Dunstan-Drumm', 'Euron-Greyjoy', 'Victarion-Greyjoy', 'Aerys-II-Targaryen', 'Robert-Baratheon', 'Tywin-Lannister', 'Alla-Tyrell', 'Elinor-Tyrell', 'Margaery-Tyrell', 'Megga-Tyrell', 'Osney-Kettleblack', 'Armen', 'Leo-Tyrell', 'Marwyn', 'Mollander', 'Pate-(novice)', 'Roone', 'Jon-Arryn', 'Amerei-Frey', 'Lancel-Lannister', 'Andrey-Dalt', 'Arianne-Martell', 'Garin-(orphan)', 'Myrcella-Baratheon', 'Sylva-Santagar', 'Tyene-Sand', 'Anya-Waynwood', 'Benedar-Belmore', 'Gilwood-Hunter', 'Harrold-Hardyng', 'Horton-Redfort', 'Petyr-Baelish', 'Symond-Templeton', 'Yohn-Royce', 'Areo-Hotah', 'Arys-Oakheart', 'Caleotte', 'Doran-Martell', 'Obara-Sand', 'Oberyn-Martell', 'Gerold-Dayne', 'Arya-Stark', 'Sandor-Clegane', 'Sansa-Stark', 'Rodrik-Harlaw', 'Aurane-Waters', 'Gyles-Rosby', 'Harys-Swyft', 'Loras-Tyrell', 'Orton-Merryweather', 'Stannis-Baratheon', 'Tommen-Baratheon', 'Balman-Byrch', 'Bronn', 'Falyse-Stokeworth', 'Balon-Swann', 'Gregor-Clegane', 'Beric-Dondarrion', 'Catelyn-Stark', 'Thoros-of-Myr', 'Brienne-of-Tarth', 'Blue-Bard', 'Lambert-Turnberry', 'Qyburn', 'Tallad', 'Boros-Blount', 'Meryn-Trant', 'Osmund-Kettleblack', 'Bran-Stark', 'Robb-Stark', 'Dick-Crabb', 'Dontos-Hollard', 'Elder-Brother', 'Hyle-Hunt', 'Joffrey-Baratheon', 'Lysa-Arryn', 'Meribald', 'Narbert', 'Podrick-Payne', 'Pyg', 'Randyll-Tarly', 'Renly-Baratheon', 'Shagwell', 'Timeon', 'Tyrion-Lannister',

'Daven-Lannister', 'Edmure-Tully', 'Edwyn-Frey', 'Emmon-Frey', 'Ryman-Frey', 'Dorcus', 'Dunsen', 'Eddard-Stark', 'High-Sparrow', 'Ilyn-Payne', 'Jalabhar-Xho', 'Jocelyn-Swyft', 'Kevan-Lannister', 'Mace-Tyrell', 'Maegor-I-Targaryen', 'Moon-Boy', 'Olenna-Redwyne', 'Osfryd-Kettleblack', 'Pycelle', 'Rafford', 'Taena-of-Myr', 'Varys', 'Walder-Frey', 'Colemon', 'Gretchen', 'Maddy', 'Robert-Arryn', 'Craster', 'Nymeria-Sand', 'Walder-Rivers', 'Genna-Lannister', 'Hotho-Harlaw', 'Nute', 'Pia', 'Lothor-Bruno', 'Mya-Stone', 'Nestor-Royce', 'Myranda-Royce']

4-Core: ['Mace-Tyrell', 'Robert-Arryn', 'Walder-Frey', 'Jocelyn-Swyft', 'Olenna-Redwyne', 'Nestor-Royce', 'Gyles-Rosby', 'Orton-Merryweather', 'Dunsen', 'Dontos-Hollard', 'Rhaegar-Targaryen', 'Aurane-Waters', 'Harys-Swyft', 'Tommen-Baratheon', 'Eddard-Stark', 'Colemon', 'Meryn-Trant', 'Brienne-of-Tarth', 'Taena-of-Myr', 'Osmund-Kettleblack', 'Kevan-Lannister', 'Ryman-Frey', 'Cersei-Lannister', 'Margaery-Tyrell', 'Pate-(novice)', 'Robb-Stark', 'Qyburn', 'Doran-Martell', 'Emmon-Frey', 'Tyene-Sand', 'Joffrey-Baratheon', 'Yohn-Royce', 'Varys', 'Lancel-Lannister', 'Gretchen', 'Edmure-Tully', 'Brynden-Tully', 'Jon-Snow', 'Euron-Greyjoy', 'Armen', 'Randall-Tarly', 'Lyle-Crakehall', 'Garin-(orphan)', 'Rafford', 'Boros-Blount', 'Arianne-Martell', 'Arys-Oakheart', 'Loras-Tyrell', 'Ilyn-Payne', 'Tywin-Lannister', 'Pycelle', 'Osney-Kettleblack', 'Dorcus', 'Asha-Greyjoy', 'Leo-Tyrell', 'Petyr-Baelish', 'Renly-Baratheon', 'Hyle-Hunt', 'Mollander', 'Sansa-Stark', 'Myrcella-Baratheon', 'Aerys-II-Targaryen', 'Anya-Waynwood', 'Sylva-Santagar', 'Symond-Templeton', 'Andrey-Dalt', 'Gerold-Dayne', 'Gregor-Clegane', 'Victarion-Greyjoy', 'Roone', 'Aero-Hotah', 'Oberry-Martell', 'Baelor-Blacktyde', 'Gilwood-Hunter', 'Jon-Arryn', 'Addam-Marbrand', 'Catelyn-Stark', 'Samwell-Tarly', 'Tyrion-Lannister', 'Horton-Redfort', 'Sandor-Clegane', 'Balon-Greyjoy', 'Lysa-Arryn', 'Obara-Sand', 'High-Sparrow', 'Alleras', 'Benedar-Belmore', 'Nymeria-Sand', 'Stannis-Baratheon', 'Jaime-Lannister', 'Aeron-Greyjoy', 'Robert-Baratheon', 'Maddy']

5-Core: ['Mace-Tyrell', 'Tyene-Sand', 'Joffrey-Baratheon', 'Sandor-Clegane', 'Petyr-Baelish', 'Renly-Baratheon', 'Sansa-Stark', 'Myrcella-Baratheon', 'High-Sparrow', 'Jocelyn-Swyft', 'Gyles-Rosby', 'Orton-Merryweather', 'Dunsen', 'Aurane-Waters', 'Sylva-Santagar', 'Andrey-Dalt', 'Harys-Swyft', 'Tommen-Baratheon', 'Eddard-Stark', 'Gregor-Clegane', 'Meryn-Trant', 'Garin-(orphan)', 'Rafford', 'Boros-Blount', 'Arianne-Martell', 'Brienne-of-Tarth', 'Loras-Tyrell', 'Taena-of-Myr', 'Osmund-Kettleblack', 'Ilyn-Payne', 'Kevan-Lannister', 'Stannis-Baratheon', 'Cersei-Lannister', 'Jaime-Lannister', 'Margaery-Tyrell', 'Tywin-Lannister', 'Qyburn', 'Pycelle', 'Catelyn-Stark', 'Robert-Baratheon', 'Doran-Martell', 'Osney-Kettleblack', 'Tyrion-Lannister', 'Dorcus']

6-Core: ['Mace-Tyrell', 'Joffrey-Baratheon', 'Sandor-Clegane', 'Petyr-Baelish', 'Renly-Baratheon', 'Sansa-Stark', 'Gyles-Rosby', 'Orton-Merryweather', 'Aurane-Waters', 'Harys-Swyft', 'Tommen-Baratheon', 'Eddard-Stark', 'Gregor-Clegane', 'Brienne-of-Tarth', 'Loras-Tyrell', 'Taena-of-Myr', 'Osmund-Kettleblack', 'Kevan-Lannister', 'Stannis-Baratheon', 'Cersei-Lannister', 'Jaime-Lannister', 'Margaery-Tyrell', 'Tywin-Lannister', 'Qyburn', 'Pycelle', 'Catelyn-Stark', 'Robert-Baratheon', 'Osney-Kettleblack', 'Tyrion-Lannister']

7-Core: ['Mace-Tyrell', 'Joffrey-Baratheon', 'Gyles-Rosby', 'Orton-Merryweather', 'Aurane-Waters', 'Harys-Swyft', 'Tommen-Baratheon', 'Taena-of-Myr', 'Loras-Tyrell', 'Osmund-Kettleblack', 'Stannis-Baratheon', 'Cersei-

Lannister', 'Jaime-Lannister', 'Margaery-Tyrell', 'Tywin-Lannister', 'Qyburn', 'Pycelle', 'Robert-Baratheon', 'Tyrion-Lannister']

8-Core: []
9-Core: []
10-Core: []
11-Core: []
12-Core: []
13-Core: []
14-Core: []

All found K-Cores in Book 5

2-Core: ['Daenerys-Targaryen', 'Aegon-Targaryen-(son-of-Rhaegar)', 'Elia-Martell', 'Franklyn-Flowers', 'Haldon', 'Harry-Strickland', 'Jon-Connington', 'Lemore', 'Rhaegar-Targaryen', 'Rolly-Duckfield', 'Tyrion-Lannister', 'Tywin-Lannister', 'Viserys-Targaryen', 'Yandry', 'Ysilla', 'Aemon-Targaryen-(Maester-Aemon)', 'Clydas', 'Gilly', 'Jon-Snow', 'Samwell-Tarly', 'Stannis-Baratheon', 'Aenys-Frey', 'Hosteen-Frey', 'Theon-Greyjoy', 'Wyman-Manderly', 'Aeron-Greyjoy', 'Euron-Greyjoy', 'Victarion-Greyjoy', 'Aerys-II-Targaryen', 'Robert-Baratheon', 'Aggo', 'Barristan-Selmy', 'Daario-Naharis', 'Grey-Worm', 'Jhogo', 'Rakharo', 'Alys-Karstark', 'Melisandre', 'Sigorn', 'Alysane-Mormont', 'Asha-Greyjoy', 'Justin-Massey', 'Archibald-Yronwood', 'Cletus-Yronwood', 'Gerris-Drinkwater', 'Quentyn-Martell', 'Tattered-Prince', 'Areo-Hotah', 'Arianne-Martell', 'Arys-Oakheart', 'Balon-Swann', 'Doran-Martell', 'Ellaria-Sand', 'Obara-Sand', 'Myrcella-Baratheon', 'Arnolf-Karstark', 'Mors-Umber', 'Arron', 'Emrick', 'Hareth-(Moles-Town)', 'Satin', 'Arya-Stark', 'Bran-Stark', 'Eddard-Stark', 'Jeyne-Poole', 'Kindly-Man', 'Mance-Rayder', 'Ramsay-Snow', 'Sansa-Stark', 'Waif', 'Gerold-Dayne', 'Clayton-Suggs', 'Godry-Farring', 'Grimtongue', 'Hagen', 'Qarl-the-Maid', 'Sybelle-Glover', 'Tristifer-Botley', 'Tycho-Nestoris', 'Axell-Florent', 'Selyse-Florent', 'Shireen-Baratheon', 'Balon-Greyjoy', 'Barbrey-Dustin', 'Roger-Ryswell', 'Roose-Bolton', 'Belwas', 'Ben-Plumm', 'Bloodbeard', 'Galazza-Galare', 'Grazhar', 'Grolio', 'Hizdahr-zo-Loraq', 'Khazz', 'Missandei', 'Reznak-mo-Reznak', 'Skahaz-mo-Kandaq', 'Steelskin', 'Varys', 'Xaro-Xhoan-Daxos', 'Davos-Seaworth', 'Belaquo', 'Goghor-the-Giant', 'Irri', 'Spotted-Cat', 'Ben-Bones', 'Kasporio', 'Penny', 'Tybero-Istarion', 'Benerro', 'Moqorro', 'Benethon-Scales', 'Narbert-Grandison', 'Boros-Blount', 'Meryn-Trant', 'Tommen-Baratheon', 'Bowen-Marsh', 'Cellador', 'Eddison-Tollett', 'Othell-Yarwyck', 'Tormund', 'Wick-Whittlestick', 'Brynden-Rivers', 'Coldhands', 'Hodor', 'Jojen-Reed', 'Leaf', 'Meera-Reed', 'Rickon-Stark', 'Robb-Stark', 'Brienne-of-Tarth', 'Cersei-Lannister', 'Jaime-Lannister', 'Brus-Buckler', 'Malegorn', 'Caggo', 'Meris', 'Caleotte', 'Gregor-Clegane', 'Dunsen', 'High-Sparrow', 'Ilyn-Payne', 'Joffrey-Baratheon', 'Kevan-Lannister', 'Mace-Tyrell', 'Moelle', 'Qyburn', 'Rafford', 'Robert-Strong', 'Scolera', 'Unella', 'Cleon', 'Kedry', 'Mully', 'Cotter-Pyke', 'Denys-Mallister', 'Cromm', 'Hero', 'Jhiqui', 'Drogo', 'Illyrio-Mopatis', 'Jorah-Mormont', 'Marselen', 'Mezzara', 'Miklaz', 'Qavo-Nogarys', 'Qezza', 'Symon-Stripeback', 'Dalla', 'Val', 'Damon-Dance-for-Me', 'Grunt', 'Skinner', 'Sour-Alyn', 'Garth-(Wolfs-Den)', 'Leona-Woolfield', 'Marlon-Manderly', 'Robett-Glover', 'Salladhor-Saan', 'Sybell-Spicer', 'Devan-Seaworth', 'Donal-Noye', 'Qhorin-Halfhand', 'Nymeria-Sand', 'Oberyn-Martell',

'Willam-Dustin', 'Emmett', 'Hobb', 'Janos-Slynt', 'Hop-Robin', 'Frenya',
'Holly', 'Fulk', 'Garth-Greyfeather', 'Hairy-Hal', 'Jack-Bulwer', 'Grenn',
'Pypar', 'Todder', 'Lorren', 'Leathers', 'Rory', 'Myles-Toyne', 'Pycelle',
'Owen', 'Rowan', 'Hother-Umber', 'Jeor-Mormont', 'Squirrel', 'Patrek-of-Kings-
Mountain', 'Rattleshirt', 'Toregg', 'Weeper', 'Wun-Weg-Wun-Dar-Wun', 'Ygritte',
'Nurse', 'Joramun', 'Richard-Horpe', 'Randyll-Tarly', 'Rodrik-Sparr', 'Yezzan-
zo-Qaggaz', 'Tyene-Sand', 'Patchface', 'Sweets', 'Walder-Frey-(son-of-Merrett)',
'Yellow-Dick', 'Rodrik-Cassel', 'Walda-Frey-(daughter-of-Merrett)', 'Scar',
'Theomore', 'Walder-Frey-(son-of-Jammes)']

3-Core: ['Daenerys-Targaryen', 'Aegon-Targaryen-(son-of-Rhaegar)', 'Haldon',
'Jon-Connington', 'Lemore', 'Rhaegar-Targaryen', 'Rolly-Duckfield', 'Tyrion-
Lannister', 'Tywin-Lannister', 'Viserys-Targaryen', 'Yandry', 'Ysilla', 'Aemon-
Targaryen-(Maester-Aemon)', 'Clydas', 'Gilly', 'Jon-Snow', 'Samwell-Tarly',
'Stannis-Baratheon', 'Theon-Greyjoy', 'Wyman-Manderly', 'Aerys-II-Targaryen',
'Robert-Baratheon', 'Aggo', 'Barristan-Selmy', 'Daario-Naharis', 'Grey-Worm',
'Jhogo', 'Rakharo', 'Alys-Karstark', 'Melisandre', 'Sigorn', 'Alysane-Mormont',
'Asha-Greyjoy', 'Justin-Massey', 'Archibald-Yronwood', 'Cletus-Yronwood',
'Gerris-Drinkwater', 'Quentyn-Martell', 'Tattered-Prince', 'Areo-Hotah',
'Arianne-Martell', 'Balon-Swann', 'Doran-Martell', 'Ellaria-Sand', 'Obara-Sand',
'Myrcella-Baratheon', 'Arnolf-Karstark', 'Mors-Umber', 'Arron', 'Emrick',
'Hareth-(Moles-Town)', 'Satin', 'Arya-Stark', 'Bran-Stark', 'Eddard-Stark',
'Jeyne-Poole', 'Mance-Rayder', 'Ramsay-Snow', 'Sansa-Stark', 'Clayton-Suggs',
'Godry-Farring', 'Hagen', 'Qarl-the-Maid', 'Tristifer-Botley', 'Tycho-Nestoris',
'Axell-Florent', 'Selyse-Florent', 'Shireen-Baratheon', 'Barbrey-Dustin',
'Roose-Bolton', 'Belwas', 'Ben-Plumm', 'Bloodbeard', 'Galazza-Galare',
'Grazhar', 'Grolio', 'Hizdahr-zo-Loraq', 'Khazz', 'Missandei', 'Reznak-mo-
Reznak', 'Skahaz-mo-Kandaq', 'SteelSkin', 'Varys', 'Davos-Seaworth', 'Belaquo',
'Goghor-the-Giant', 'Irri', 'Spotted-Cat', 'Kasporio', 'Penny', 'Tybero-
Istarion', 'Benerro', 'Moqorro', 'Narbert-Grandison', 'Meryn-Trant', 'Tommen-
Baratheon', 'Bowen-Marsh', 'Cellador', 'Eddison-Tollett', 'Othell-Yarwyck',
'Tormund', 'Coldhands', 'Hodor', 'Jojen-Reed', 'Meera-Reed', 'Rickon-Stark',
'Robb-Stark', 'Cersei-Lannister', 'Jaime-Lannister', 'Brus-Buckler', 'Malegorn',
'Caggo', 'Meris', 'Gregor-Clegane', 'Dunsen', 'High-Sparrow', 'Ilyn-Payne',
'Joffrey-Baratheon', 'Kevan-Lannister', 'Mace-Tyrell', 'Moelle', 'Rafford',
'Robert-Strong', 'Scolera', 'Unella', 'Mully', 'Hero', 'Jhiqui', 'Illyrio-
Mopatis', 'Jorah-Mormont', 'Qavo-Nogarys', 'Qezza', 'Dalla', 'Val', 'Damon-
Dance-for-Me', 'Grunt', 'Skinner', 'Sour-Alyn', 'Devan-Seaworth', 'Emmett',
'Hobb', 'Janos-Slynt', 'Hop-Robin', 'Holly', 'Garth-Greyfeather', 'Hairy-Hal',
'Jack-Bulwer', 'Leathers', 'Rowan', 'Hother-Umber', 'Squirrel', 'Patrek-of-
King-Mountain', 'Toregg', 'Weeper', 'Wun-Weg-Wun-Dar-Wun', 'Nurse', 'Randyll-
Tarly', 'Yezzan-zo-Qaggaz', 'Sweets', 'Yellow-Dick']

4-Core: ['Skahaz-mo-Kandaq', 'Hizdahr-zo-Loraq', 'Mully', 'Belwas', 'Godry-
Farring', 'Meris', 'Goghor-the-Giant', 'Khazz', 'Yandry', 'Irri', 'Sour-Alyn',
'Barbrey-Dustin', 'Barristan-Selmy', 'Dunsen', 'Rhaegar-Targaryen', 'Tommen-
Baratheon', 'Eddard-Stark', 'Meryn-Trant', 'Mance-Rayder', 'Kevan-Lannister',
'Rakharo', 'Missandei', 'Cersei-Lannister', 'Robb-Stark', 'Bloodbeard', 'Davos-
Seaworth', 'Bowen-Marsh', 'Clydas', 'Eddison-Tollett', 'Doran-Martell',
'Haldon', 'Viserys-Targaryen', 'Hero', 'Varys', 'Jhiqui', 'Ysilla', 'Val',

'Tattered-Prince', 'Shireen-Baratheon', 'Aemon-Targaryen-(Maester-Aemon)',
 'Squirrel', 'Jon-Snow', 'Hobb', 'Jon-Connington', 'Lemore', 'Roose-Bolton',
 'Daenerys-Targaryen', 'Jhogo', 'Arya-Stark', 'Rafford', 'Spotted-Cat', 'Illyn-
 Payne', 'Tywin-Lannister', 'Quentyn-Martell', 'Asha-Greyjoy', 'Theon-Greyjoy',
 'Gilly', 'Bran-Stark', 'Damon-Dance-for-Me', 'Selyse-Florent', 'Myrcella-
 Baratheon', 'Ben-Plumm', 'Rolly-Duckfield', 'Grey-Worm', 'Axell-Florent',
 'Daario-Naharis', 'Gregor-Clegane', 'Aggo', 'Tormund', 'Illyrio-Mopatis',
 'Reznak-mo-Reznak', 'Gerris-Drinkwater', 'Cletus-Yronwood', 'Samwell-Tarly',
 'Grolio', 'Melisandre', 'Tyrion-Lannister', 'Archibald-Yronwood', 'Satin',
 'Skinner', 'Dalla', 'Aegon-Targaryen-(son-of-Rhaegar)', 'Caggo', 'Belaquo',
 'Galazza-Galare', 'Ramsay-Snow', 'Jeyne-Poole', 'Stannis-Baratheon', 'Jaime-
 Lannister', 'Wyman-Manderly', 'Robert-Baratheon', 'Rowan']
 5-Core: ['Haldon', 'Skahaz-mo-Kandaq', 'Hizdahr-zo-Loraq', 'Theon-Greyjoy',
 'Belwas', 'Jhiqui', 'Bran-Stark', 'Ysilla', 'Val', 'Selyse-Florent', 'Shireen-
 Baratheon', 'Yandry', 'Rolly-Duckfield', 'Ben-Plumm', 'Irri', 'Aegon-
 Targaryen-(son-of-Rhaegar)', 'Barbrey-Dustin', 'Grey-Worm', 'Jon-Snow', 'Jon-
 Connington', 'Barristan-Selmy', 'Axell-Florent', 'Lemore', 'Roose-Bolton',
 'Dunsen', 'Rhaegar-Targaryen', 'Daario-Naharis', 'Daenerys-Targaryen', 'Eddard-
 Stark', 'Galazza-Galare', 'Aggo', 'Meryn-Trant', 'Jhogo', 'Arya-Stark', 'Mance-
 Rayder', 'Gregor-Clegane', 'Rafford', 'Illyrio-Mopatis', 'Ramsay-Snow', 'Reznak-
 mo-Reznak', 'Jeyne-Poole', 'Illyn-Payne', 'Stannis-Baratheon', 'Rakharo',
 'Missandei', 'Cersei-Lannister', 'Robb-Stark', 'Tywin-Lannister', 'Quentyn-
 Martell', 'Wyman-Manderly', 'Robert-Baratheon', 'Davos-Seaworth', 'Doran-
 Martell', 'Grolio', 'Melisandre', 'Tyrion-Lannister']
 6-Core: ['Haldon', 'Skahaz-mo-Kandaq', 'Missandei', 'Hizdahr-zo-Loraq', 'Daario-
 Naharis', 'Daenerys-Targaryen', 'Yandry', 'Rolly-Duckfield', 'Irri', 'Aegon-
 Targaryen-(son-of-Rhaegar)', 'Ysilla', 'Grey-Worm', 'Jhiqui', 'Jon-Connington',
 'Barristan-Selmy', 'Tyrion-Lannister', 'Lemore']
 7-Core: []
 8-Core: []
 9-Core: []
 10-Core: []
 11-Core: []
 12-Core: []
 13-Core: []
 14-Core: []

All found K-Cores in All Books

2-Core: ['Addam-Marbrand', 'Jaime-Lannister', 'Tywin-Lannister', 'Aegon-I-
 Targaryen', 'Daenerys-Targaryen', 'Eddard-Stark', 'Aemon-Targaryen-(Maester-
 Aemon)', 'Alliser-Thorne', 'Bowen-Marsh', 'Chett', 'Clydas', 'Jeor-Mormont',
 'Jon-Snow', 'Samwell-Tarly', 'Aerys-II-Targaryen', 'Brandon-Stark', 'Gerold-
 Hightower', 'Jon-Arryn', 'Robert-Baratheon', 'Aggo', 'Drogo', 'Jhogo', 'Jorah-
 Mormont', 'Quaro', 'Rakharo', 'Albett', 'Halder', 'Rast', 'Grenn', 'Pypar',
 'Tyrion-Lannister', 'Alyn', 'Harwin', 'Jory-Cassel', 'Tomard', 'Arthur-Dayne',
 'Arya-Stark', 'Benjen-Stark', 'Bran-Stark', 'Catelyn-Stark', 'Cersei-Lannister',
 'Desmond', 'Illyn-Payne', 'Jeyne-Poole', 'Joffrey-Baratheon', 'Meryn-Trant',
 'Mordane', 'Mycah', 'Myrcella-Baratheon', 'Petyr-Baelish', 'Rickon-Stark',

'Robb-Stark', 'Rodrik-Cassel', 'Sandor-Clegane', 'Sansa-Stark', 'Syrio-Forel', 'Tommen-Baratheon', 'Vayon-Poole', 'Yoren', 'Arys-Oakheart', 'Balon-Greyjoy', 'Balon-Swann', 'Renly-Baratheon', 'Barristan-Selmy', 'Boros-Blount', 'Pycelle', 'Varys', 'Jaremy-Rykker', 'Luwin', 'Mance-Rayder', 'Theon-Greyjoy', 'Waymar-Royce', 'Beric-Dondarrion', 'Gregor-Clegane', 'Loras-Tyrell', 'Thoros-of-Myr', 'Hallis-Mollen', 'Hodor', 'Hullen', 'Joseth', 'Nan', 'Osha', 'Rickard-Karstark', 'Rickard-Stark', 'Stiv', 'Lyanna-Stark', 'Bronn', 'Chiggen', 'Marillion', 'Shae', 'Shagga', 'Vardis-Egen', 'Willis-Wode', 'Brynden-Tully', 'Edmure-Tully', 'Hoster-Tully', 'Lysa-Arryn', 'Nestor-Royce', 'Walder-Frey', 'Colemon', 'Eon-Hunter', 'Jon-Umber-(Greatjon)', 'Mya-Stone', 'Robert-Arryn', 'Stevron-Frey', 'Tytos-Blackwood', 'Wendel-Manderly', 'Cayn', 'Janos-Slynt', 'Stannis-Baratheon', 'Chella', 'Clement-Piper', 'Karyl-Vance', 'Cohollo', 'Haggo', 'Qotho', 'Conn', 'Doreah', 'Illyrio-Mopatis', 'Irri', 'Jhiqui', 'Mirri-Maz-Dur', 'Rhaegar-Targaryen', 'Viserys-Targaryen', 'Danwell-Frey', 'Hosteen-Frey', 'Jared-Frey', 'Dareon', 'Daryn-Hornwood', 'Torrhen-Karstark', 'Donal-Noye', 'Jommo', 'Ogo', 'Dywen', 'Galbart-Glover', 'Gendry', 'High-Septon-(fat_one)', 'Howland-Reed', 'Marq-Piper', 'Raymun-Darry', 'Wyl-(guard)', 'Roose-Bolton', 'Gared', 'Will-(prologue)', 'Oswell-Whent', 'Todder', 'Harys-Swyft', 'Heward', 'Hobb', 'Kevan-Lannister', 'Maege-Mormont', 'Jonos-Bracken', 'Jyck', 'Morrec', 'Lancel-Lannister', 'Mace-Tyrell', 'Lyn-Corbray', 'Paxter-Redwyne', 'Maegor-I-Targaryen', 'Randyll-Tarly', 'Timett', 'Harren-Hoare', 'Rhaenys-Targaryen', 'Visenya-Targaryen', 'Daeron-II-Targaryen', 'Aenys-Frey', 'Aeron-Greyjoy', 'Dagmer', 'Aggar', 'Gynir', 'Ramsay-Snow', 'Belwas', 'Alayaya', 'Alebelly', 'Hayhead', 'Jojen-Reed', 'Mikken', 'Alester-Florent', 'Cortnay-Penrose', 'Davos-Seaworth', 'Allar-Deem', 'Allard-Seaworth', 'Marya-Seaworth', 'Amabel', 'Harra', 'Amory-Lorch', 'Chiswyck', 'Dunsen', 'Hot-Pie', 'Pollicher', 'Rafford', 'Tickler', 'Vargo-Hoat', 'Monford-Velaryon', 'Biter', 'Jaen-Hghar', 'Koss', 'Lommy-Greenhands', 'Rorge', 'Weasel', 'Weese', 'Woth', 'Asha-Greyjoy', 'Wex-Pyke', 'Axell-Florent', 'Selyse-Florent', 'Ygritte', 'Ballabar', 'Podrick-Payne', 'Victarion-Greyjoy', 'Mandon-Moore', 'Osmund-Kettleblack', 'Benfred-Tallhart', 'Helman-Tallhart', 'Leobald-Tallhart', 'Stygg', 'Beth-Cassel', 'Gyles-Rosby', 'Tanda-Stokeworth', 'Chayle', 'Cley-Cerwyn', 'Donella-Hornwood', 'Meera-Reed', 'Reek', 'Stafford-Lannister', 'Walder-Frey-(son-of-Jammos)', 'Walder-Frey-(son-of-Merrett)', 'Wyman-Manderly', 'Brienne-of-Tarth', 'Emmon-Cuy', 'Robar-Royce', 'Vyman', 'Jacelyn-Bywater', 'Bryce-Caron', 'Guyard-Morrigen', 'Cleos-Frey', 'Colen-of-Greenpools', 'Desmond-Grell', 'Margaery-Tyrell', 'Mathis-Rowan', 'Utherydes-Wayn', 'Osfryd-Kettleblack', 'Osney-Kettleblack', 'Preston-Greenfield', 'Vylarr', 'Craster', 'Eddison-Tollett', 'Thoren-Smallwood', 'Cressen', 'Melisandre', 'Patchface', 'Pylos', 'Shireen-Baratheon', 'Pono', 'Pyat-Pree', 'Xaro-Xhoan-Daxos', 'Dalbridge', 'Ebbens', 'Qhorin-Halfhand', 'Devan-Seaworth', 'Salladhor-Saan', 'Mors-Umber', 'Dontos-Hollard', 'Moon-Boy', 'Doran-Martell', 'Drennan', 'Squint', 'Stonesnake', 'Euron-Greyjoy', 'Garlan-Tyrell', 'Lollys-Stokeworth', 'Hallyne', 'High-Septon-(Tyrions)', 'Tysha', 'Jarman-Buckwell', 'Jason-Mallister', 'Ottyn-Wythers', 'Lark', 'Rattleshirt', 'Lorren', 'Urzen', 'Trystane-Martell', 'Olyvar-Frey', 'Palla', 'Perwyn-Frey', 'Qyburn', 'Steffon-Baratheon', 'Robett-Glover', 'Werlag', 'Jalabhar-Xho', 'Oberyn-Martell', 'Aegon-Frey-(son-of-Stevron)', 'Torrhen-Stark', 'Aegon-Targaryen-(son-of-Rhaegar)', 'Elia-Martell', 'Cotter-

Pyke', 'Denys-Mallister', 'Owen', 'Alerie-Hightower', 'Lamprey', 'Alla-Tyrell', 'Elinor-Tyrell', 'Megga-Tyrell', 'Anders-Yronwood', 'Andrew-Estermont', 'Edric-Storm', 'Anguy', 'Greenbeard', 'Husband', 'Lem', 'Sharna', 'Tom-of-Sevenstreams', 'Arstan', 'Kraznys-mo-Nakloz', 'Mero', 'Lewyn-Martell', 'Elmar-Frey', 'Porridge', 'Baelor-I-Targaryen', 'Bannen', 'Bedwyck', 'Ben-Plumm', 'Oznak-zo-Pahl', 'Daario-Naharis', 'Jack-Bulwer', 'Big-Boil', 'Othell-Yarwyck', 'Coldhands', 'Brella', 'Walton', 'Butterbumps', 'Olenna-Redwyne', 'Dacey-Mormont', 'Jeyne-Westerling', 'Lothar-Frey', 'Raynald-Westerling', 'Robin-Ryger', 'Roslin-Frey', 'Ryman-Frey', 'Tansy', 'Ellaria-Sand', 'Willas-Tyrell', 'Small-Paul', 'Cleon', 'Ghael', 'Clubfoot-Karl', 'Gilly', 'Dirk', 'Tormund', 'Prendahl-na-Ghezn', 'Sallor', 'Grey-Worm', 'Grolio', 'Missandei', 'Dalla', 'Val', 'Guncer-Sunglass', 'Khorane-Sathmantes', 'Dick-Follard', 'Satin', 'Mag-Mar-Tun-Doh-Weg', 'Lothor-Brone', 'Errok', 'Grigg', 'Styr', 'Gawen-Westerling', 'Sybell-Spicer', 'Gendel', 'Gorne', 'Kegs', 'Jarl', 'Hareth-(Moles-Town)', 'Mully', 'Spare-Boot', 'Harma', 'Varamyr', 'Weeper', 'Shagwell', 'Orell', 'Ryk', 'Jon-Umber-(Smalljon)', 'Robin-Flint', 'Jonothor-Darry', 'Joramun', 'Willem-Lannister', 'Symon-Silver-Tongue', 'Merrett-Frey', 'Walder-Rivers', 'Martyn-Lannister', 'Oswell-Kettleblack', 'Yohn-Royce', 'Petyr-Frey', 'Rollam-Westerling', 'Rhaella-Targaryen', 'Zollo', 'Lyle-Crakehall', 'Alleras', 'Baelor-Blacktyde', 'Dunstan-Drumm', 'Gorold-Goodbrother', 'Meldred-Merlyn', 'Murenmore', 'Armen', 'Leo-Tyrell', 'Marwyn', 'Mollander', 'Pate-(novice)', 'Roone', 'Amerei-Frey', 'Quentyn-Martell', 'Andrey-Dalt', 'Arianne-Martell', 'Garin-(orphan)', 'Sylva-Santagar', 'Tyene-Sand', 'Anya-Waynwood', 'Benedar-Belmore', 'Gilwood-Hunter', 'Harrold-Hardyng', 'Horton-Redfort', 'Symond-Templeton', 'Areo-Hotah', 'Caleotte', 'Obara-Sand', 'Cedra', 'Gerold-Dayne', 'Mellei', 'Morra', 'Brea', 'Brusco', 'Denyo-Terys', 'Kindly-Man', 'Talea', 'Ternesio-Terys', 'Waif', 'Red-Oarsman', 'Rodrik-Harlaw', 'Tristifer-Botley', 'Aurane-Waters', 'Orton-Merryweather', 'Balman-Byrch', 'Falyse-Stokeworth', 'Mariya-Darry', 'Blue-Bard', 'Lambert-Turnberry', 'Tallad', 'Creighton-Longbough', 'Dick-Crabb', 'Elder-Brother', 'Hyle-Hunt', 'Illifer', 'Meribald', 'Narbert', 'Pyg', 'Timeon', 'Willow-Heddle', 'Daven-Lannister', 'Edwyn-Frey', 'Emmon-Frey', 'Dorcus', 'High-Sparrow', 'Jocelyn-Swyft', 'Maggy', 'Melara-Hetherspoon', 'Senelle', 'Taena-of-Myr', 'Unella', 'Gretchel', 'Maddy', 'Shadrich', 'Daemon-Sand', 'Nymeria-Sand', 'Forley-Prester', 'Genna-Lannister', 'Hotho-Harlaw', 'Humfrey-Hewett', 'Nute', 'Quellon-Greyjoy', 'Terrance-Lynderly', 'Kojja-Mo', 'Gyles-Grafton', 'Sigfryd-Harlaw', 'Josmyn-Peckledon', 'Pia', 'Ronnet-Connington', 'Wylis-Manderly', 'Lyonel-Corbray', 'Myranda-Royce', 'Ralf-Stonehouse', 'Quhuru-Mo', 'Xhondo', 'Franklyn-Flowers', 'Haldon', 'Harry-Strickland', 'Jon-Connington', 'Lemore', 'Rolly-Duckfield', 'Yandry', 'Ysilla', 'Alys-Karstark', 'Sigorn', 'Alysane-Mormont', 'Justin-Massey', 'Archibald-Yronwood', 'Cletus-Yronwood', 'Gerris-Drinkwater', 'Tattered-Prince', 'Arnolf-Karstark', 'Arron', 'Emrick', 'Clayton-Suggs', 'Godry-Farring', 'Grimtongue', 'Hagen', 'Qarl-the-Maid', 'Sybelle-Glover', 'Tycho-Nestoris', 'Barbrey-Dustin', 'Roger-Ryswell', 'Bloodbeard', 'Galazza-Galare', 'Grazhar', 'Hizdahr-zo-Loraq', 'Khazz', 'Reznak-mo-Reznak', 'Skahaz-mo-Kandaq', 'Steelskin', 'Belaquo', 'Goghor-the-Giant', 'Spotted-Cat', 'Ben-Bones', 'Kasporio', 'Penny', 'Tybero-Istarion', 'Benerro', 'Moqorro', 'Benethon-Scales', 'Narbert-Grandison', 'Cellador', 'Wick-Whittlestick', 'Brynden-Rivers', 'Leaf', 'Brus-Buckler',

'Malegorn', 'Caggo', 'Meris', 'Moelle', 'Robert-Strong', 'Scolera', 'Kedry',
'Cromm', 'Hero', 'Marselen', 'Mezzara', 'Miklaz', 'Qavo-Nogarys', 'Qezza',
'Symon-Stripeback', 'Damon-Dance-for-Me', 'Grunt', 'Skinner', 'Sour-Alyn',
'Garth-(Wolfs-Den)', 'Leona-Woolfield', 'Marlon-Manderly', 'Willam-Dustin',
'Emmett', 'Hop-Robin', 'Frenya', 'Holly', 'Fulk', 'Garth-Greyfeather', 'Hairy-
Hal', 'Leathers', 'Rory', 'Myles-Toyne', 'Rowan', 'Hother-Umber', 'Rhaegar-
Frey', 'Squirrel', 'Patrek-of-Kings-Mountain', 'Toregg', 'Wun-Weg-Wun-Dar-Wun',
'Nurse', 'Richard-Horpe', 'Rodrik-Sparr', 'Yezzan-zo-Qaggaz', 'Sweets', 'Yellow-
Dick', 'Walda-Frey-(daughter-of-Merrett)', 'Scar', 'Theomore']

3-Core: ['Addam-Marbrand', 'Jaime-Lannister', 'Tywin-Lannister', 'Aegon-I-
Targaryen', 'Daenerys-Targaryen', 'Eddard-Stark', 'Aemon-Targaryen-(Maester-
Aemon)', 'Alliser-Thorne', 'Bowen-Marsh', 'Chett', 'Clydas', 'Jeor-Mormont',
'Jon-Snow', 'Samwell-Tarly', 'Aerys-II-Targaryen', 'Brandon-Stark', 'Gerold-
Hightower', 'Jon-Arryn', 'Robert-Baratheon', 'Aggo', 'Drogo', 'Jhogo', 'Jorah-
Mormont', 'Quaro', 'Rakharo', 'Albett', 'Halder', 'Rast', 'Grenn', 'Pypar',
'Tyrion-Lannister', 'Alyn', 'Harwin', 'Jory-Cassel', 'Tomard', 'Arthur-Dayne',
'Arya-Stark', 'Benjen-Stark', 'Bran-Stark', 'Catelyn-Stark', 'Cersei-Lannister',
'Illyn-Payne', 'Jeyne-Poole', 'Joffrey-Baratheon', 'Meryn-Trant', 'Mordane',
'Mycah', 'Myrcella-Baratheon', 'Petyr-Baelish', 'Rickon-Stark', 'Robb-Stark',
'Rodrik-Cassel', 'Sandor-Clegane', 'Sansa-Stark', 'Syrio-Forel', 'Tommen-
Baratheon', 'Vayon-Poole', 'Yoren', 'Arys-Oakheart', 'Balon-Greyjoy', 'Balon-
Swann', 'Renly-Baratheon', 'Baristan-Selmy', 'Boros-Blount', 'Pycelle',
'Varys', 'Jaremy-Rykker', 'Luwin', 'Mance-Rayder', 'Theon-Greyjoy', 'Beric-
Dondarrion', 'Gregor-Clegane', 'Loras-Tyrell', 'Thoros-of-Myr', 'Hallis-Mollen',
'Hodor', 'Hullen', 'Nan', 'Osha', 'Rickard-Karstark', 'Lyanna-Stark', 'Bronn',
'Chiggen', 'Marillion', 'Shae', 'Shagga', 'Vardis-Egen', 'Willis-Wode',
'Brynden-Tully', 'Edmure-Tully', 'Hoster-Tully', 'Lysa-Arryn', 'Nestor-Royce',
'Walder-Frey', 'Colemon', 'Eon-Hunter', 'Jon-Umber-(Greatjon)', 'Mya-Stone',
'Robert-Arryn', 'Stevron-Frey', 'Tytos-Blackwood', 'Wendel-Manderly', 'Cayn',
'Janos-Slynt', 'Stannis-Baratheon', 'Chella', 'Karyl-Vance', 'Cohollo', 'Haggo',
'Qotho', 'Doreah', 'Illyrio-Mopatis', 'Irri', 'Jhiqui', 'Mirri-Maz-Duur',
'Rhaegar-Targaryen', 'Viserys-Targaryen', 'Dareon', 'Donal-Noye', 'Dywen',
'Galbart-Glover', 'Gendry', 'High-Septon-(fat_one)', 'Marq-Piper',
'Wyl-(guard)', 'Roose-Bolton', 'Todder', 'Harys-Swyft', 'Hobb', 'Kevan-
Lannister', 'Maege-Mormont', 'Lancel-Lannister', 'Mace-Tyrell', 'Lyn-Corbray',
'Paxter-Redwyne', 'Maegor-I-Targaryen', 'Randyll-Tarly', 'Timett', 'Harren-
Hoare', 'Aenys-Frey', 'Aeron-Greyjoy', 'Dagmer', 'Ramsay-Snow', 'Belwas',
'Alayaya', 'Alebelly', 'Jojen-Reed', 'Mikken', 'Alester-Florent', 'Cortnay-
Penrose', 'Davos-Seaworth', 'Amory-Lorch', 'Chiswyck', 'Dunsen', 'Hot-Pie',
'Polliver', 'Rafford', 'Tickler', 'Vargo-Hoat', 'Biter', 'Jaqen-Hghar', 'Lommy-
Greenhands', 'Rorge', 'Weasel', 'Weese', 'Woth', 'Asha-Greyjoy', 'Wex-Pyke',
'Axell-Florent', 'Selyse-Florent', 'Ygritte', 'Podrick-Payne', 'Victarion-
Greyjoy', 'Mandon-Moore', 'Osmund-Kettleblack', 'Benfred-Tallhart', 'Helman-
Tallhart', 'Leobald-Tallhart', 'Beth-Cassel', 'Gyles-Rosby', 'Tanda-Stokeworth',
'Cley-Cerwyn', 'Donella-Hornwood', 'Meera-Reed', 'Stafford-Lannister', 'Walder-
Frey-(son-of-Jammes)', 'Walder-Frey-(son-of-Merrett)', 'Wyman-Manderly',
'Brienne-of-Tarth', 'Emmon-Cuy', 'Robar-Royce', 'Vyman', 'Jacelyn-Bywater',
'Bryce-Caron', 'Cleos-Frey', 'Desmond-Grell', 'Margaery-Tyrell', 'Mathis-Rowan',

'Utherydes-Wayn', 'Osfryd-Kettleblack', 'Osney-Kettleblack', 'Craster',
'Eddison-Tollett', 'Thoren-Smallwood', 'Cressen', 'Melisandre', 'Patchface',
'Pylos', 'Shireen-Baratheon', 'Pono', 'Dalbridge', 'Ebben', 'Qhorin-Halfhand',
'Devan-Seaworth', 'Salladhor-Saan', 'Mors-Umber', 'Dontos-Hollard', 'Moon-Boy',
'Doran-Martell', 'Stonesnake', 'Euron-Greyjoy', 'Garlan-Tyrell', 'Lollys-
Stokeworth', 'High-Septon-(Tyrions)', 'Tysha', 'Jarman-Buckwell', 'Jason-
Mallister', 'Ottyn-Wythers', 'Lark', 'Rattleshirt', 'Lorren', 'Qyburn', 'Robett-
Glover', 'Jalabhar-Xho', 'Oberyn-Martell', 'Aegon-Frey-(son-of-Stevron)',
'Aegon-Targaryen-(son-of-Rhaegar)', 'Elia-Martell', 'Cotter-Pyke', 'Denys-
Mallister', 'Owen', 'Lamprey', 'Alla-Tyrell', 'Elinor-Tyrell', 'Megga-Tyrell',
'Edric-Storm', 'Anguy', 'Greenbeard', 'Lem', 'Sharna', 'Tom-of-Sevenstreams',
'Arstan', 'Mero', 'Elmar-Frey', 'Porridge', 'Baelor-I-Targaryen', 'Bedwyck',
'Ben-Plumm', 'Daario-Naharis', 'Jack-Bulwer', 'Othell-Yarwyck', 'Coldhands',
'Walton', 'Olenna-Redwyne', 'Dacey-Mormont', 'Jeyne-Westerling', 'Lothar-Frey',
'Roslin-Frey', 'Ryman-Frey', 'Tansy', 'Ellaria-Sand', 'Willas-Tyrell', 'Small-
Paul', 'Gilly', 'Tormund', 'Grey-Worm', 'Grolio', 'Missandei', 'Dalla', 'Val',
'Khorane-Sathmantes', 'Satin', 'Lothor-Brune', 'Errok', 'Grigg', 'Styr',
'Sybell-Spicer', 'Kegs', 'Jarl', 'Hareth-(Moles-Town)', 'Mully', 'Spare-Boot',
'Harma', 'Varamyr', 'Weeper', 'Shagwell', 'Ryk', 'Robin-Flint', 'Symon-Silver-
Tongue', 'Walder-Rivers', 'Yohn-Royce', 'Zollo', 'Lyle-Crakehall', 'Alleras',
'Baelor-Blacktyde', 'Dunstan-Drumm', 'Armen', 'Leo-Tyrell', 'Marwyn',
'Mollander', 'Pate-(novice)', 'Roone', 'Amerei-Frey', 'Quentyn-Martell',
'Andrey-Dalt', 'Arianne-Martell', 'Garin-(orphan)', 'Sylva-Santagar', 'Tyene-
Sand', 'Anya-Waynwood', 'Benedar-Belmore', 'Gilwood-Hunter', 'Harrold-Hardyng',
'Horton-Redfort', 'Symond-Templeton', 'Areo-Hotah', 'Caleotte', 'Obara-Sand',
'Gerold-Dayne', 'Rodrik-Harlaw', 'Tristifer-Botley', 'Aurane-Waters', 'Orton-
Merryweather', 'Balman-Byrch', 'Falyse-Stokeworth', 'Blue-Bard', 'Lambert-
Turnberry', 'Tallad', 'Dick-Crabb', 'Elder-Brother', 'Hyle-Hunt', 'Meribald',
'Narbert', 'Pyg', 'Timeon', 'Daven-Lannister', 'Edwyn-Frey', 'Emmon-Frey',
'Dorcus', 'High-Sparrow', 'Jocelyn-Swyft', 'Taena-of-Myr', 'Unella', 'Gretchel',
'Maddy', 'Nymeria-Sand', 'Genna-Lannister', 'Hotho-Harlaw', 'Nute', 'Pia',
'Myranda-Royce', 'Haldon', 'Jon-Connington', 'Lemore', 'Rolly-Duckfield',
'Yandry', 'Ysilla', 'Alys-Karstark', 'Sigorn', 'Alysane-Mormont', 'Justin-
Massey', 'Archibald-Yronwood', 'Cletus-Yronwood', 'Gerris-Drinkwater',
'Tattered-Prince', 'Arnolf-Karstark', 'Arron', 'Emrick', 'Clayton-Suggs',
'Godry-Farring', 'Grimtongue', 'Hagen', 'Qarl-the-Maid', 'Tycho-Nestoris',
'Barbrey-Dustin', 'Bloodbeard', 'Galazza-Galare', 'Grazhar', 'Hizdahr-zo-Loraq',
'Khazz', 'Reznak-mo-Reznak', 'Skahaz-mo-Kandaq', 'SteelSkin', 'Belaquo',
'Goghor-the-Giant', 'Spotted-Cat', 'Kasporio', 'Penny', 'Tybero-Istarion',
'Benerro', 'Moqorro', 'Narbert-Grandison', 'Cellador', 'Brus-Buckler',
'Malegorn', 'Caggo', 'Meris', 'Moelle', 'Robert-Strong', 'Scolera', 'Hero',
'Qavo-Nogarys', 'Qezza', 'Damon-Dance-for-Me', 'Grunt', 'Skinner', 'Sour-Alyn',
'Emmett', 'Hop-Robin', 'Holly', 'Garth-Greyfeather', 'Hairy-Hal', 'Leathers',
'Rowan', 'Hother-Umber', 'Squirrel', 'Patrek-of-Kings-Mountain', 'Toregg', 'Wun-
Weg-Wun-Dar-Wun', 'Nurse', 'Yezzan-zo-Qaggaz', 'Sweets', 'Yellow-Dick']
4-Core: ['Beric-Dondarrion', 'Willas-Tyrell', 'Hizdahr-zo-Loraq', 'Greenbeard',
'Robert-Arryn', 'Drogo', 'Rorge', 'Godry-Farring', 'Walder-Frey', 'Yandry',
'Jaremy-Rykker', 'Olenna-Redwyne', 'Ottyn-Wythers', 'Lyanna-Stark', 'Dunsen',

'Dontos-Hollard', 'Rhaegar-Targaryen', 'Aurane-Waters', 'Taena-of-Myr', 'Ryman-Frey', 'Rakharo', 'Rickard-Karstark', 'Bronn', 'Ygritte', 'Jhiqui', 'Edmure-Tully', 'Jon-Snow', 'Owen', 'Daenerys-Targaryen', 'Garin-(orphan)', 'Arys-Oakheart', 'Loras-Tyrell', 'Spotted-Cat', 'Marillion', 'Quentyn-Martell', 'Weasel', 'Doreah', 'Wendel-Manderly', 'Yoren', 'Cleos-Frey', 'Stafford-Lannister', 'Craster', 'Mirri-Maz-Duur', 'Damon-Dance-for-Me', 'Hyle-Hunt', 'Mollander', 'Sansa-Stark', 'Vardis-Egen', 'Ben-Plumm', 'Maegor-I-Targaryen', 'Walton', 'Kegs', 'Polliver', 'Sylva-Santagar', 'Daario-Naharis', 'Maege-Mormont', 'Baelor-Blacktyde', 'Lothar-Frey', 'Garlan-Tyrell', 'Gilwood-Hunter', 'Cletus-Yronwood', 'Catelyn-Stark', 'Sybell-Spicer', 'Cohollo', 'Qhorin-Halfhand', 'Sandor-Clegane', 'Bedwyck', 'Brandon-Stark', 'Lysa-Arryn', 'Ellaria-Sand', 'Tysha', 'Obara-Sand', 'Aegon-Targaryen-(son-of-Rhaegar)', 'Megga-Tyrell', 'Caggo', 'Thoros-of-Myr', 'Jorah-Mormont', 'Cortnay-Penrose', 'Sharna', 'Harma', 'Robert-Baratheon', 'Vargo-Hoat', 'Lark', 'Rickon-Stark', 'Hullen', 'Robett-Glover', 'Meris', 'Jocelyn-Swyft', 'Barbrey-Dustin', 'Nestor-Royce', 'Barristan-Selmy', 'Gyles-Rosby', 'Eddard-Stark', 'Cressen', 'Dareon', 'Small-Paul', 'Kevan-Lannister', 'Alyn', 'Cersei-Lannister', 'Margaery-Tyrell', 'Bowen-Marsh', 'Caleotte', 'Viserys-Targaryen', 'Penny', 'Hero', 'Halder', 'Willis-Wode', 'Lancel-Lannister', 'Val', 'Tattered-Prince', 'Anguy', 'Brynden-Tully', 'Randyll-Tarly', 'Elia-Martell', 'Weese', 'Arya-Stark', 'Woth', 'Arianne-Martell', 'Hother-Umber', 'Ilyn-Payne', 'Balon-Swann', 'Timett', 'Donal-Noye', 'Edric-Storm', 'Bran-Stark', 'Petyr-Baelish', 'Leo-Tyrell', 'Renly-Baratheon', 'Biter', 'Benjen-Stark', 'Selyse-Florent', 'Myrcella-Baratheon', 'Grey-Worm', 'Salladhor-Saan', 'Tormund', 'Reznak-mo-Reznak', 'Shagga', 'Arstan', 'Moqorro', 'Lothor-Brune', 'Samwell-Tarly', 'Groleo', 'Chett', 'Jaqen-Hghar', 'Roslin-Frey', 'Jon-Umber-(Greatjon)', 'Benerro', 'High-Septon-(Tyrions)', 'Belaquo', 'Cotter-Pyke', 'Benedar-Belmore', 'Styr', 'Ramsay-Snow', 'Nymeria-Sand', 'Alester-Florent', 'Mandon-Moore', 'Harwin', 'Maddy', 'Janos-Slynt', 'Amory-Lorch', 'Skahaz-mo-Kandaq', 'Moon-Boy', 'Hallis-Mollen', 'Gendry', 'Mathis-Rowan', 'Goghor-the-Giant', 'Tanda-Stokeworth', 'Orton-Merryweather', 'Meera-Reed', 'Harys-Swyft', 'Meryn-Trant', 'Mance-Rayder', 'Gerold-Hightower', 'Ryk', 'Osmund-Kettleblack', 'Walder-Frey-(son-of-Merrett)', 'Jojen-Reed', 'Pate-(novice)', 'Tomard', 'Qyburn', 'Bloodbeard', 'Davos-Seaworth', 'Doran-Martell', 'Eddison-Tollett', 'Haldon', 'Tyene-Sand', 'Joffrey-Baratheon', 'Shireen-Baratheon', 'Aemon-Targaryen-(Maester-Aemon)', 'Squirrel', 'Pypar', 'Rast', 'Walder-Rivers', 'Hobb', 'Podrick-Payne', 'Robert-Strong', 'Thoren-Smallwood', 'Lemore', 'Lyle-Crakehall', 'Stevron-Frey', 'Boros-Blount', 'Tywin-Lannister', 'Syrio-Forel', 'Osney-Kettleblack', 'Tickler', 'Asha-Greyjoy', 'Gilly', 'Shae', 'Spare-Boot', 'Rolly-Duckfield', 'Dywen', 'Weeper', 'Elinor-Tyrell', 'Elmar-Frey', 'Hareth-(Moles-Town)', 'Axell-Florent', 'Pono', 'Aerys-II-Targaryen', 'Rodrik-Cassel', 'Anya-Waynwood', 'Andrey-Dalt', 'Gerold-Dayne', 'Gregor-Clegane', 'Lollys-Stokeworth', 'Mya-Stone', 'Aleo-Hotah', 'Oberyn-Martell', 'Gerris-Drinkwater', 'Hoster-Tully', 'Jon-Arryn', 'Rattleshirt', 'Lem', 'Tyrion-Lannister', 'Archibald-Yronwood', 'Qotho', 'Jalabhar-Xho', 'Horton-Redfort', 'Satin', 'Dick-Crabb', 'Skinner', 'Dalla', 'Blue-Bard', 'Vayon-Poole', 'Alleras', 'Timeon', 'Aegon-Frey-(son-of-Stevron)', 'Mycah', 'Lommy-Greenhands', 'Tansy', 'Quaro', 'Stannis-Baratheon', 'Jaime-Lannister', 'Arnolf-Karstark', 'Aeron-Greyjoy', 'Marq-Piper', 'Mace-Tyrell', 'Mully',

'Belwas', 'Cellador', 'Luwin', 'Tallad', 'Khrazz', 'Jeor-Mormont', 'Irri',
'Sour-Alyn', 'Patchface', 'Dagmer', 'Tommen-Baratheon', 'Colemon', 'Haggo',
'Galbart-Glover', 'Brienne-of-Tarth', 'Missandei', 'Nan', 'Robb-Stark',
'Clydas', 'Emmon-Frey', 'Yohn-Royce', 'Varys', 'Ysilla', 'Gretchen', 'Jarl',
'Jacelyn-Bywater', 'Tom-of-Sevenstreams', 'Euron-Greyjoy', 'Armen', 'Jon-
Connington', 'Osha', 'Roose-Bolton', 'Mors-Umber', 'Coldhands', 'Donella-
Hornwood', 'Alliser-Thorne', 'Jhogo', 'Rafford', 'Pyg', 'Todder', 'Pycelle',
'Emmett', 'Dorcus', 'Theon-Greyjoy', 'Pylos', 'Zollo', 'Shagwell', 'Grenn',
'Aegon-I-Targaryen', 'Jeyne-Westerling', 'Symond-Templeton', 'Aggo', 'Victarion-
Greyjoy', 'Illyrio-Mopatis', 'Roone', 'Hodor', 'Edwyn-Frey', 'Addam-Marbrand',
'Melisandre', 'Devan-Seaworth', 'Chiswyck', 'Denys-Mallister', 'Mordane',
'Balon-Greyjoy', 'Paxter-Redwyne', 'High-Sparrow', 'Alla-Tyrell', 'Othell-
Yarwyck', 'Galazza-Galare', 'Hot-Pie', 'Jory-Cassel', 'Jeyne-Poole', 'Wyman-
Manderly', 'Rowan']

5-Core: ['Amory-Lorch', 'Beric-Dondarrion', 'Marq-Piper', 'Rickon-Stark',
'Hullen', 'Lark', 'Mace-Tyrell', 'Willas-Tyrell', 'Hizdahr-zo-Loraq', 'Skahaz-
mo-Kandaq', 'Robett-Glover', 'Mully', 'Greenbeard', 'Robert-Arryn', 'Drogo',
'Belwas', 'Rorge', 'Godry-Farring', 'Luwin', 'Hallis-Mollen', 'Gendry', 'Walder-
Frey', 'Mathis-Rowan', 'Jeor-Mormont', 'Irri', 'Yandry', 'Olenna-Redwyne',
'Jocelyn-Swyft', 'Nestor-Royce', 'Barbrey-Dustin', 'Tanda-Stokeworth',
'Barristan-Selmy', 'Gyles-Rosby', 'Orton-Merryweather', 'Lyanna-Stark',
'Dunsen', 'Dagmer', 'Rhaegar-Targaryen', 'Meera-Reed', 'Dontos-Hollard',
'Aurane-Waters', 'Tommen-Baratheon', 'Harys-Swyft', 'Eddard-Stark', 'Haggo',
'Meryn-Trant', 'Galbart-Glover', 'Mance-Rayder', 'Gerold-Hightower', 'Brienne-
of-Tarth', 'Cressen', 'Taena-of-Myr', 'Osmund-Kettleblack', 'Dareon', 'Small-
Paul', 'Kevan-Lannister', 'Ryman-Frey', 'Rakharo', 'Alyn', 'Cersei-Lannister',
'Nan', 'Rickard-Karstark', 'Robb-Stark', 'Jojen-Reed', 'Tomard', 'Walder-
Frey-(son-of-Merrett)', 'Margaery-Tyrell', 'Qyburn', 'Missandei', 'Bowen-Marsh',
'Davos-Seaworth', 'Clydas', 'Eddison-Tollett', 'Doran-Martell', 'Emmon-Frey',
'Haldon', 'Viserys-Targaryen', 'Tyene-Sand', 'Joffrey-Baratheon', 'Bronn',
'Ygritte', 'Halder', 'Varys', 'Jhiqui', 'Willis-Wode', 'Lancel-Lannister',
'Ysilla', 'Val', 'Anguy', 'Jarl', 'Edmure-Tully', 'Shireen-Baratheon', 'Aemon-
Targaryen-(Maester-Aemon)', 'Brynden-Tully', 'Pypar', 'Jacelyn-Bywater', 'Rast',
'Jon-Snow', 'Euron-Greyjoy', 'Tom-of-Sevenstreams', 'Hobb', 'Jon-Connington',
'Podrick-Payne', 'Owen', 'Thoren-Smallwood', 'Osha', 'Roose-Bolton', 'Randyll-
Tarly', 'Lemore', 'Elia-Martell', 'Weese', 'Daenerys-Targaryen', 'Alliser-
Thorne', 'Jhogo', 'Arya-Stark', 'Rafford', 'Boros-Blount', 'Woth', 'Arianne-
Martell', 'Arys-Oakheart', 'Garin-(orphan)', 'Loras-Tyrell', 'Ilyn-Payne',
'Marillion', 'Balon-Swann', 'Tywin-Lannister', 'Pycelle', 'Quentyn-Martell',
'Doreah', 'Donal-Noye', 'Edric-Storm', 'Osney-Kettleblack', 'Tickler', 'Dorcus',
'Yoren', 'Cleos-Frey', 'Asha-Greyjoy', 'Theon-Greyjoy', 'Pylos', 'Gilly',
'Craster', 'Bran-Stark', 'Petyr-Baelish', 'Mirri-Maz-Duur', 'Renly-Baratheon',
'Shae', 'Hyle-Hunt', 'Sansa-Stark', 'Benjen-Stark', 'Biter', 'Myrcella-
Baratheon', 'Grenn', 'Selyse-Florent', 'Spare-Boot', 'Ben-Plumm', 'Rolly-
Duckfield', 'Aegon-I-Targaryen', 'Dywen', 'Maegor-I-Targaryen', 'Grey-Worm',
'Weeper', 'Elmar-Frey', 'Hareth-(Moles-Town)', 'Axell-Florent', 'Aerys-II-
Targaryen', 'Rodrik-Cassel', 'Kegs', 'Polliver', 'Sylva-Santagar', 'Daario-
Naharis', 'Andrey-Dalt', 'Gerold-Dayne', 'Gregor-Clegane', 'Aggo', 'Tormund',

'Victarion-Greyjoy', 'Illyrio-Mopatis', 'Lollys-Stokeworth', 'Reznak-mo-Reznak', 'Aleo-Hotah', 'Oberyn-Martell', 'Hodor', 'Lothar-Frey', 'Hoster-Tully', 'Garlan-Tyrell', 'Arstan', 'Jon-Arryn', 'Addam-Marbrand', 'Catelyn-Stark', 'Samwell-Tarly', 'Rattleshirt', 'Lem', 'Groleo', 'Melisandre', 'Tyrion-Lannister', 'Chett', 'Devan-Seaworth', 'Jaqen-Hghar', 'Chiswyck', 'Qotho', 'Qhorin-Halfhand', 'Sandor-Clegane', 'Denys-Mallister', 'Bedwyck', 'Mordane', 'Satin', 'Brandon-Stark', 'Balon-Greyjoy', 'Lysa-Arryn', 'Paxter-Redwyne', 'Ellaria-Sand', 'Roslin-Frey', 'Jon-Umber-(Greatjon)', 'High-Sparrow', 'Dalla', 'Aegon-Targaryen-(son-of-Rhaegar)', 'Vayon-Poole', 'Cotter-Pyke', 'Thoros-of-Myr', 'Othell-Yarwyck', 'Galazza-Galare', 'Jorah-Mormont', 'Styr', 'Hot-Pie', 'Ramsay-Snow', 'Cortnay-Penrose', 'Mycah', 'Jory-Cassel', 'Jeyne-Poole', 'Lommy-Greenhands', 'Tansy', 'Quaro', 'Stannis-Baratheon', 'Alester-Florent', 'Mandon-Moore', 'Jaime-Lannister', 'Harma', 'Harwin', 'Wyman-Manderly', 'Robert-Baratheon', 'Aeron-Greyjoy', 'Vargo-Hoat', 'Janos-Slynt']

6-Core: ['Amory-Lorch', 'Beric-Dondarrion', 'Marq-Piper', 'Rickon-Stark', 'Hullen', 'Willas-Tyrell', 'Mace-Tyrell', 'Hizdahr-zo-Loraq', 'Skahaz-mo-Kandaq', 'Mully', 'Greenbeard', 'Robert-Arryn', 'Drogo', 'Belwas', 'Luwin', 'Gendry', 'Walder-Frey', 'Mathis-Rowan', 'Jeor-Mormont', 'Irri', 'Yandry', 'Olenna-Redwyne', 'Nestor-Royce', 'Tanda-Stokeworth', 'Barristan-Selmy', 'Gyles-Rosby', 'Orton-Merryweather', 'Dunsen', 'Dontos-Hollard', 'Rhaegar-Targaryen', 'Meera-Reed', 'Aurane-Waters', 'Tommen-Baratheon', 'Harys-Swyft', 'Eddard-Stark', 'Meryn-Trant', 'Galbart-Glover', 'Mance-Rayder', 'Brienne-of-Tarth', 'Cressen', 'Taena-of-Myr', 'Osmund-Kettleblack', 'Kevan-Lannister', 'Ryman-Frey', 'Rakharo', 'Cersei-Lannister', 'Nan', 'Jojen-Reed', 'Rickard-Karstark', 'Robb-Stark', 'Walder-Frey-(son-of-Merrett)', 'Margaery-Tyrell', 'Missandei', 'Qyburn', 'Bowen-Marsh', 'Davos-Seaworth', 'Clydas', 'Eddison-Tollett', 'Doran-Martell', 'Haldon', 'Viserys-Targaryen', 'Joffrey-Baratheon', 'Bronn', 'Ygritte', 'Varys', 'Jhiqui', 'Lancel-Lannister', 'Ysilla', 'Val', 'Anguy', 'Jarl', 'Edmure-Tully', 'Shireen-Baratheon', 'Aemon-Targaryen-(Maester-Aemon)', 'Brynden-Tully', 'Pypar', 'Jacelyn-Bywater', 'Tom-of-Sevenstreams', 'Jon-Snow', 'Hobb', 'Jon-Connington', 'Podrick-Payne', 'Owen', 'Osha', 'Roose-Bolton', 'Randyll-Tarly', 'Lemore', 'Elia-Martell', 'Weese', 'Daenerys-Targaryen', 'Alliser-Thorne', 'Jhogo', 'Arya-Stark', 'Rafford', 'Boros-Blount', 'Arianne-Martell', 'Arys-Oakheart', 'Loras-Tyrell', 'Ilyn-Payne', 'Marillion', 'Balon-Swann', 'Tywin-Lannister', 'Pycelle', 'Quentyn-Martell', 'Doreah', 'Donal-Noye', 'Osney-Kettleblack', 'Tickler', 'Yoren', 'Cleos-Frey', 'Theon-Greyjoy', 'Gilly', 'Craster', 'Bran-Stark', 'Petyr-Baelish', 'Renly-Baratheon', 'Shae', 'Hyle-Hunt', 'Sansa-Stark', 'Benjen-Stark', 'Selyse-Florent', 'Myrcella-Baratheon', 'Grenn', 'Ben-Plumm', 'Rolly-Duckfield', 'Dywen', 'Grey-Worm', 'Elmar-Frey', 'Axell-Florent', 'Aerys-II-Targaryen', 'Rodrik-Cassel', 'Polliver', 'Daario-Naharis', 'Gregor-Clegane', 'Aggo', 'Tormund', 'Illyrio-Mopatis', 'Lollys-Stokeworth', 'Aleo-Hotah', 'Oberyn-Martell', 'Hodor', 'Lothar-Frey', 'Hoster-Tully', 'Garlan-Tyrell', 'Arstan', 'Jon-Arryn', 'Addam-Marbrand', 'Catelyn-Stark', 'Samwell-Tarly', 'Rattleshirt', 'Lem', 'Melisandre', 'Tyrion-Lannister', 'Chiswyck', 'Qhorin-Halfhand', 'Sandor-Clegane', 'Denys-Mallister', 'Mordane', 'Satin', 'Brandon-Stark', 'Balon-Greyjoy', 'Lysa-Arryn', 'Paxter-Redwyne', 'Ellaria-Sand', 'Jon-Umber-(Greatjon)', 'High-Sparrow', 'Dalla', 'Aegon-Targaryen-(son-of-Rhaegar)', 'Cotter-Pyke', 'Thoros-of-Myr', 'Othell-Yarwyck', 'Thornos-of-Myr', 'Othell-Yarwyck']

'Jorah-Mormont', 'Styr', 'Hot-Pie', 'Ramsay-Snow', 'Cortnay-Penrose', 'Jory-Cassel', 'Jeyne-Poole', 'Tansy', 'Alester-Florent', 'Stannis-Baratheon', 'Mandon-Moore', 'Jaime-Lannister', 'Harma', 'Harwin', 'Wyman-Manderly', 'Robert-Baratheon', 'Vargo-Hoat', 'Janos-Slynt']

7-Core: ['Amory-Lorch', 'Beric-Dondarrion', 'Marq-Piper', 'Rickon-Stark', 'Willas-Tyrell', 'Hizdahr-zo-Loraq', 'Mace-Tyrell', 'Skahaz-mo-Kandaq', 'Robert-Arryn', 'Drogo', 'Belwas', 'Luwin', 'Gendry', 'Walder-Frey', 'Jeor-Mormont', 'Irri', 'Olenna-Redwyne', 'Barristan-Selmy', 'Gyles-Rosby', 'Orton-Merryweather', 'Dunsen', 'Dontos-Hollard', 'Rhaegar-Targaryen', 'Meera-Reed', 'Aurane-Waters', 'Tommen-Baratheon', 'Harys-Swyft', 'Eddard-Stark', 'Meryn-Trant', 'Mance-Rayder', 'Brienne-of-Tarth', 'Cressen', 'Taena-of-Myr', 'Osmund-Kettleblack', 'Kevan-Lannister', 'Rakharo', 'Cersei-Lannister', 'Missandei', 'Jojen-Reed', 'Rickard-Karstark', 'Robb-Stark', 'Margaery-Tyrell', 'Qyburn', 'Bowen-Marsh', 'Davos-Seaworth', 'Doran-Martell', 'Eddison-Tollett', 'Viserys-Targaryen', 'Joffrey-Baratheon', 'Bronn', 'Varys', 'Jhiqui', 'Lancel-Lannister', 'Edmure-Tully', 'Shireen-Baratheon', 'Aemon-Targaryen-(Maester-Aemon)', 'Brynden-Tully', 'Pypar', 'Jacelyn-Bywater', 'Jon-Snow', 'Hobb', 'Jon-Connington', 'Osha', 'Roose-Bolton', 'Randyll-Tarly', 'Elia-Martell', 'Daenerys-Targaryen', 'Alliser-Thorne', 'Jhogo', 'Arya-Stark', 'Rafford', 'Boros-Blount', 'Loras-Tyrell', 'Ilyn-Payne', 'Marillion', 'Balon-Swann', 'Tywin-Lannister', 'Pycelle', 'Quentyn-Martell', 'Osney-Kettleblack', 'Tickler', 'Yoren', 'Cleos-Frey', 'Theon-Greyjoy', 'Gilly', 'Craster', 'Bran-Stark', 'Petyr-Baelish', 'Renly-Baratheon', 'Shae', 'Sansa-Stark', 'Benjen-Stark', 'Selyse-Florent', 'Myrcella-Baratheon', 'Grenn', 'Grey-Worm', 'Elmar-Frey', 'Aerys-II-Targaryen', 'Rodrik-Cassel', 'Polliver', 'Daario-Naharis', 'Gregor-Clegane', 'Aggo', 'Illyrio-Mopatis', 'Oberyn-Martell', 'Hodor', 'Lothar-Frey', 'Hoster-Tully', 'Jon-Arryn', 'Addam-Marbrand', 'Catelyn-Stark', 'Samwell-Tarly', 'Lem', 'Melisandre', 'Tyrion-Lannister', 'Chiswyck', 'Sandor-Clegane', 'Denys-Mallister', 'Mordane', 'Balon-Greyjoy', 'Lysa-Arryn', 'Jon-Umber-(Greatjon)', 'Aegon-Targaryen-(son-of-Rhaegar)', 'Cotter-Pyke', 'Thoros-of-Myr', 'Othell-Yarwyck', 'Jorah-Mormont', 'Ramsay-Snow', 'Jory-Cassel', 'Jeyne-Poole', 'Mandon-Moore', 'Stannis-Baratheon', 'Jaime-Lannister', 'Harwin', 'Wyman-Manderly', 'Robert-Baratheon', 'Janos-Slynt']

8-Core: ['Amory-Lorch', 'Beric-Dondarrion', 'Skahaz-mo-Kandaq', 'Rickon-Stark', 'Hizdahr-zo-Loraq', 'Mace-Tyrell', 'Robert-Arryn', 'Drogo', 'Belwas', 'Luwin', 'Gendry', 'Walder-Frey', 'Jeor-Mormont', 'Irri', 'Olenna-Redwyne', 'Barristan-Selmy', 'Gyles-Rosby', 'Orton-Merryweather', 'Dunsen', 'Rhaegar-Targaryen', 'Meera-Reed', 'Aurane-Waters', 'Tommen-Baratheon', 'Harys-Swyft', 'Eddard-Stark', 'Meryn-Trant', 'Mance-Rayder', 'Brienne-of-Tarth', 'Taena-of-Myr', 'Osmund-Kettleblack', 'Kevan-Lannister', 'Rakharo', 'Cersei-Lannister', 'Missandei', 'Jojen-Reed', 'Rickard-Karstark', 'Robb-Stark', 'Margaery-Tyrell', 'Qyburn', 'Bowen-Marsh', 'Doran-Martell', 'Eddison-Tollett', 'Viserys-Targaryen', 'Joffrey-Baratheon', 'Bronn', 'Varys', 'Jhiqui', 'Lancel-Lannister', 'Edmure-Tully', 'Aemon-Targaryen-(Maester-Aemon)', 'Brynden-Tully', 'Pypar', 'Jacelyn-Bywater', 'Jon-Snow', 'Hobb', 'Osha', 'Roose-Bolton', 'Daenerys-Targaryen', 'Alliser-Thorne', 'Jhogo', 'Arya-Stark', 'Rafford', 'Boros-Blount', 'Loras-Tyrell', 'Ilyn-Payne', 'Marillion', 'Balon-Swann', 'Tywin-Lannister', 'Pycelle', 'Tickler', 'Yoren', 'Theon-Greyjoy', 'Gilly', 'Craster', 'Bran-

Stark', 'Petyr-Baelish', 'Renly-Baratheon', 'Sansa-Stark', 'Benjen-Stark', 'Myrcella-Baratheon', 'Grenn', 'Grey-Worm', 'Elmar-Frey', 'Aerys-II-Targaryen', 'Rodrik-Cassel', 'Polliver', 'Daario-Naharis', 'Gregor-Clegane', 'Aggo', 'Illyrio-Mopatis', 'Oberyn-Martell', 'Hodor', 'Jon-Arryn', 'Addam-Marbrand', 'Catelyn-Stark', 'Samwell-Tarly', 'Melisandre', 'Tyrion-Lannister', 'Chiswyck', 'Sandor-Clegane', 'Denys-Mallister', 'Mordane', 'Lysa-Arryn', 'Jon-Umber-(Greatjon)', 'Cotter-Pyke', 'Othell-Yarwyck', 'Jorah-Mormont', 'Ramsay-Snow', 'Jory-Cassel', 'Jeyne-Poole', 'Mandon-Moore', 'Stannis-Baratheon', 'Jaime-Lannister', 'Wyman-Manderly', 'Robert-Baratheon', 'Janos-Slynt']
9-Core: ['Amory-Lorch', 'Beric-Dondarrion', 'Rickon-Stark', 'Mace-Tyrell', 'Chiswyck', 'Joffrey-Baratheon', 'Bronn', 'Janos-Slynt', 'Theon-Greyjoy', 'Varys', 'Sandor-Clegane', 'Bran-Stark', 'Mordane', 'Lancel-Lannister', 'Petyr-Baelish', 'Lysa-Arryn', 'Renly-Baratheon', 'Luwin', 'Sansa-Stark', 'Benjen-Stark', 'Walder-Frey', 'Myrcella-Baratheon', 'Edmure-Tully', 'Brynden-Tully', 'Jacelyn-Bywater', 'Jon-Snow', 'Elmar-Frey', 'Barristan-Selmy', 'Gyles-Rosby', 'Aerys-II-Targaryen', 'Rodrik-Cassel', 'Dunsen', 'Roose-Bolton', 'Polliver', 'Rhaegar-Targaryen', 'Tommen-Baratheon', 'Eddard-Stark', 'Daenerys-Targaryen', 'Gregor-Clegane', 'Meryn-Trant', 'Rafford', 'Arya-Stark', 'Boros-Blount', 'Ramsay-Snow', 'Brienne-of-Tarth', 'Loras-Tyrell', 'Jeyne-Poole', 'Jory-Cassel', 'Osmund-Kettleblack', 'Oberyn-Martell', 'Ilyn-Payne', 'Kevan-Lannister', 'Stannis-Baratheon', 'Mandon-Moore', 'Cersei-Lannister', 'Jaime-Lannister', 'Margaery-Tyrell', 'Jon-Arryn', 'Robb-Stark', 'Balon-Swann', 'Addam-Marbrand', 'Tywin-Lannister', 'Pycelle', 'Qyburn', 'Catelyn-Stark', 'Robert-Baratheon', 'Doran-Martell', 'Tyrion-Lannister', 'Tickler']
10-Core: ['Amory-Lorch', 'Beric-Dondarrion', 'Rickon-Stark', 'Mace-Tyrell', 'Chiswyck', 'Joffrey-Baratheon', 'Bronn', 'Janos-Slynt', 'Theon-Greyjoy', 'Varys', 'Sandor-Clegane', 'Bran-Stark', 'Mordane', 'Petyr-Baelish', 'Lysa-Arryn', 'Renly-Baratheon', 'Luwin', 'Sansa-Stark', 'Walder-Frey', 'Myrcella-Baratheon', 'Edmure-Tully', 'Brynden-Tully', 'Jon-Snow', 'Barristan-Selmy', 'Gyles-Rosby', 'Aerys-II-Targaryen', 'Rodrik-Cassel', 'Dunsen', 'Roose-Bolton', 'Polliver', 'Tommen-Baratheon', 'Eddard-Stark', 'Gregor-Clegane', 'Meryn-Trant', 'Rafford', 'Arya-Stark', 'Boros-Blount', 'Brienne-of-Tarth', 'Loras-Tyrell', 'Jory-Cassel', 'Osmund-Kettleblack', 'Oberyn-Martell', 'Ilyn-Payne', 'Kevan-Lannister', 'Stannis-Baratheon', 'Cersei-Lannister', 'Jaime-Lannister', 'Margaery-Tyrell', 'Jon-Arryn', 'Robb-Stark', 'Balon-Swann', 'Addam-Marbrand', 'Tywin-Lannister', 'Pycelle', 'Qyburn', 'Catelyn-Stark', 'Robert-Baratheon', 'Tyrion-Lannister', 'Tickler']
11-Core: ['Amory-Lorch', 'Mace-Tyrell', 'Chiswyck', 'Joffrey-Baratheon', 'Janos-Slynt', 'Theon-Greyjoy', 'Varys', 'Sandor-Clegane', 'Bran-Stark', 'Petyr-Baelish', 'Lysa-Arryn', 'Renly-Baratheon', 'Sansa-Stark', 'Myrcella-Baratheon', 'Jon-Snow', 'Barristan-Selmy', 'Rodrik-Cassel', 'Dunsen', 'Roose-Bolton', 'Polliver', 'Tommen-Baratheon', 'Eddard-Stark', 'Gregor-Clegane', 'Meryn-Trant', 'Rafford', 'Arya-Stark', 'Boros-Blount', 'Brienne-of-Tarth', 'Loras-Tyrell', 'Jory-Cassel', 'Osmund-Kettleblack', 'Ilyn-Payne', 'Kevan-Lannister', 'Stannis-Baratheon', 'Cersei-Lannister', 'Jaime-Lannister', 'Margaery-Tyrell', 'Jon-Arryn', 'Robb-Stark', 'Balon-Swann', 'Tywin-Lannister', 'Pycelle', 'Qyburn', 'Catelyn-Stark', 'Robert-Baratheon', 'Tyrion-Lannister', 'Tickler']
12-Core: ['Amory-Lorch', 'Mace-Tyrell', 'Chiswyck', 'Joffrey-Baratheon', 'Theon-

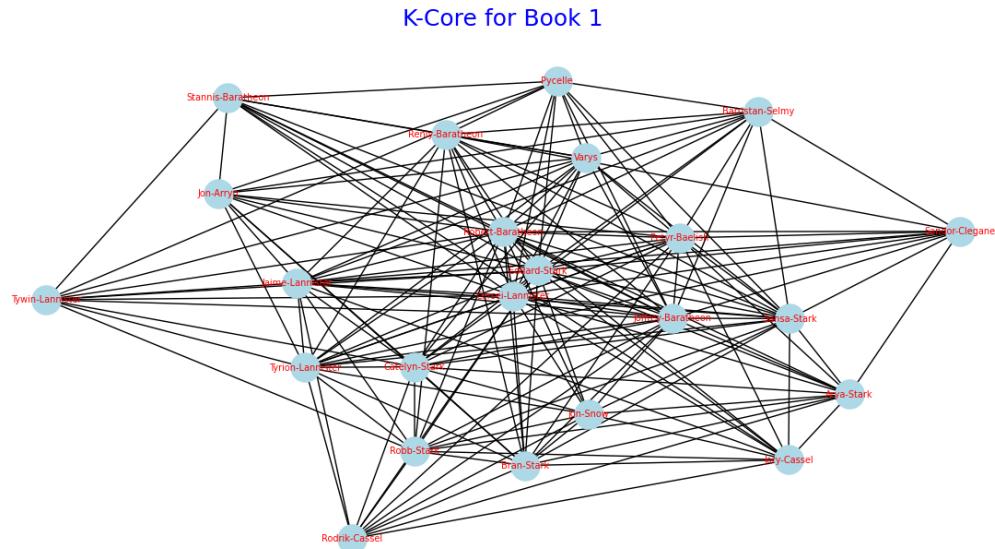
'Greyjoy', 'Varys', 'Sandor-Clegane', 'Bran-Stark', 'Petyr-Baelish', 'Lysa-Arryn', 'Renly-Baratheon', 'Sansa-Stark', 'Jon-Snow', 'Barristan-Selmy', 'Rodrik-Cassel', 'Dunsen', 'Roose-Bolton', 'Polliver', 'Tommen-Baratheon', 'Eddard-Stark', 'Gregor-Clegane', 'Meryn-Trant', 'Rafford', 'Arya-Stark', 'Brienne-of-Tarth', 'Loras-Tyrell', 'Jory-Cassel', 'Ilyn-Payne', 'Stannis-Baratheon', 'Cersei-Lannister', 'Jaime-Lannister', 'Margaery-Tyrell', 'Jon-Arryn', 'Robb-Stark', 'Tywin-Lannister', 'Pycelle', 'Catelyn-Stark', 'Robert-Baratheon', 'Tyrion-Lannister', 'Tickler']

13-Core: ['Joffrey-Baratheon', 'Varys', 'Sandor-Clegane', 'Petyr-Baelish', 'Renly-Baratheon', 'Sansa-Stark', 'Eddard-Stark', 'Gregor-Clegane', 'Arya-Stark', 'Brienne-of-Tarth', 'Loras-Tyrell', 'Stannis-Baratheon', 'Cersei-Lannister', 'Jaime-Lannister', 'Robb-Stark', 'Tywin-Lannister', 'Pycelle', 'Catelyn-Stark', 'Robert-Baratheon', 'Tyrion-Lannister']

14-Core: []

Book 1 Cores L'11-Core del libro 1 sembra identificare tutti i personaggi principali che fanno parte delle vicende di corte che accompagnano l'inizio del Trono di Spade, fino alla morte di Eddard Stark

```
[ ]: interesting_k_core_1 = nx.k_core(G_1, k=11, core_number=None)
k_core_1_nodes = plot_k_core(interesting_k_core_1, 'K-Core for Book 1', font_size=7)
```

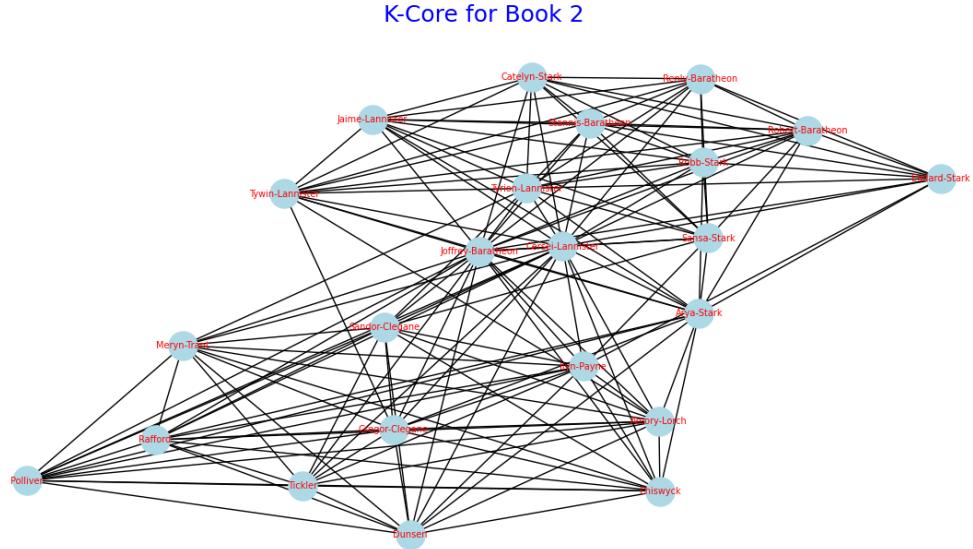


Book 2 Cores Il 10-Core del Libro 2 è interessante perché identifica due clusters.

- Il primo cluster è quello della vita di corte a castel Granito, con gli Stark e i Lannister.

- Il secondo cluster identifica la guardia reale: Tickler, Polliver, Rafford, Gregor e Sandor Clegane, Chiswyck (che avranno un ruolo centrale nel prossimi due libri).

```
[ ]: interesting_k_core_2 = nx.k_core(G_2, k=10, core_number=None)
k_core_2_nodes = plot_k_core(interesting_k_core_2, 'K-Core for Book 2', font_size=7)
```

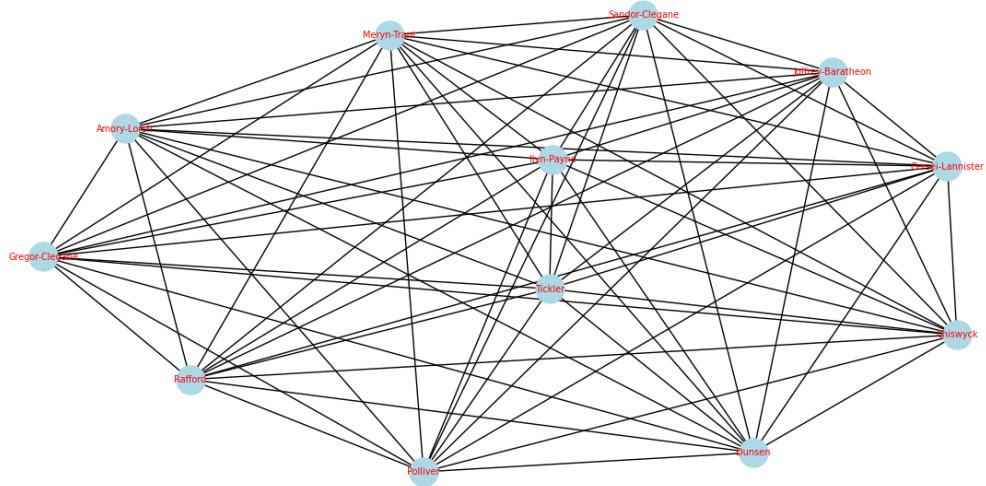


Nel Secondo Libro il K-Core più ristretto è con $K=11$. Questo Core contiene tutti i soldati al servizio dei due Clegane, che di fatto sono la guardia reale del re, ma seguono gli ordini di Cersei.

Per questo motivo il Core contiene anche Cersei-Lannister e Joffrey-Baratheon (suo figlio ed erede al trono).

```
[ ]: interesting_k_core_2 = nx.k_core(G_2, k=11, core_number=None)
k_core_2_nodes = plot_k_core(interesting_k_core_2, 'K-Core for Book 2', font_size=7)
```

K-Core for Book 2

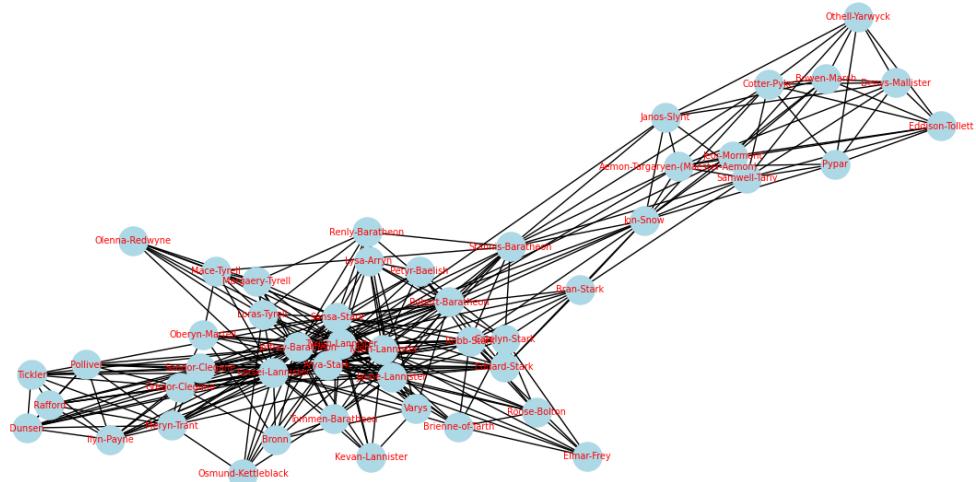


Book 3 Cores Il 7-Core del Libro 3 è molto interessante perché mostra abbastanza bene i diversi cluster di personaggi principali di tutte le storie raccontate in questo capitolo.

- Un cluster, che si distacca visibilmente dagli altri personaggi è quello della Night-Watch, che segue le vicende di Jon-Snow alla Barriera.
- Un secondo cluster, più compatto e centrale è quello dei personaggi di corte, tra gli Stark e i Lannister;
- Il terzo cluster è sempre quello della guardia reale, con i soldati che seguono gli ordini dei Clegane.

```
[ ]: interesting_k_core_3 = nx.k_core(G_3, k=7, core_number=None)
k_core_3_nodes = plot_k_core(interesting_k_core_3, 'K-Core for Book 3', □
    ↪font_size=7)
```

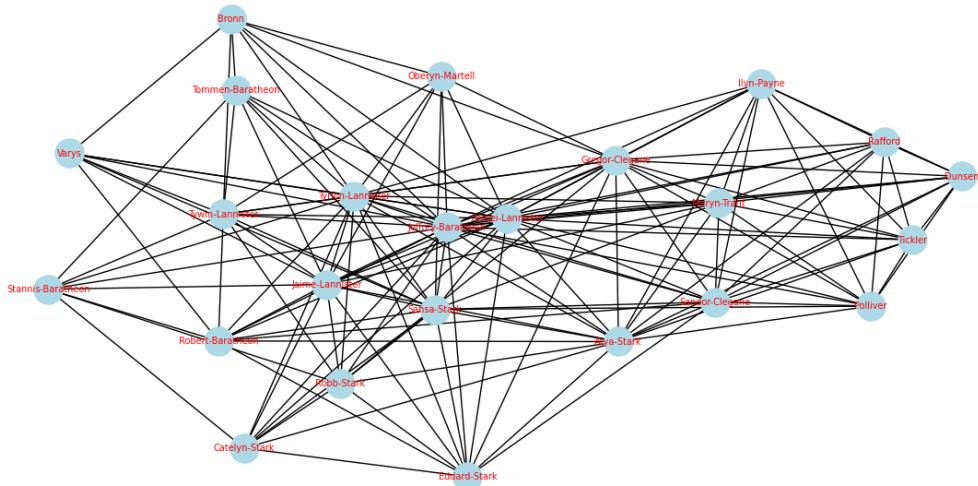
K-Core for Book 3



Stampando l'8-Core del Libro 3 evidenziamo “la vita di corte”, con i personaggi principali e il cluster della guardia reale.

```
[ ]: interesting_k_core_3 = nx.k_core(G_3, k=8, core_number=None)
k_core_3_nodes = plot_k_core(interesting_k_core_3, 'K-Core for Book 3', font_size=7)
```

K-Core for Book 3



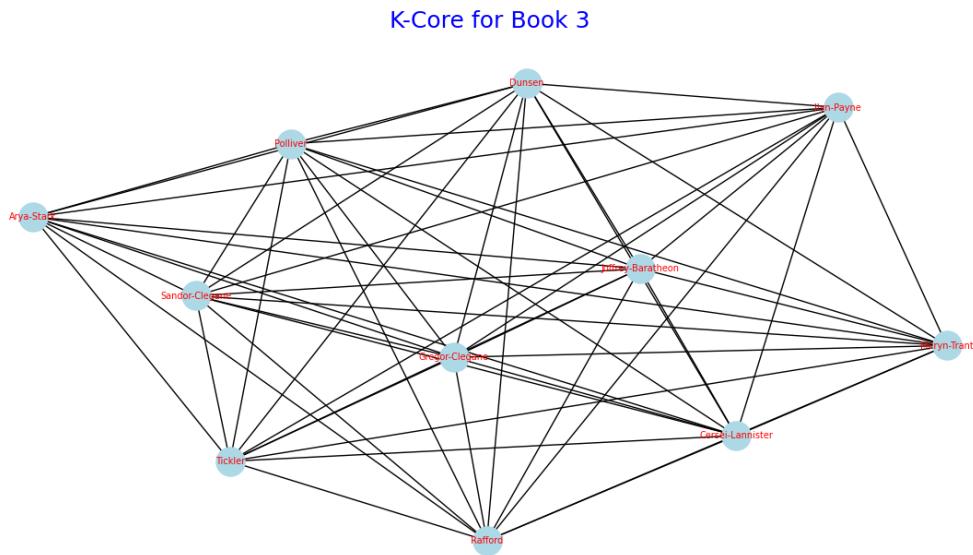
Con i Cores 9-Core e 10-Core, invece, facciamo uno zoom sulla guardia reale, su Cersie e su Joffrey, che controllano la guardia reale.

Qui compare Arya. La spiegazione è dovuta alle vicende della trama: I soldati della guardia reale saranno quelli che, dopo un grande torto nei confronti di Arya-Stark, andranno a far parte della lista di coloro che Arya promette di uccidere per vendicarsi.

Qui in pratica la guardia reale arresta il padre di Arya, che viene condannato a morte da Joffrey e da Cersei.

In pratica in questi due Cores c'è tutta la lista di quelli che Arya promette di uccidere per vendicarsi.

```
[ ]: interesting_k_core_3 = nx.k_core(G_3, k=9, core_number=None)
k_core_3_nodes = plot_k_core(interesting_k_core_3, 'K-Core for Book 3', font_size=7)
```



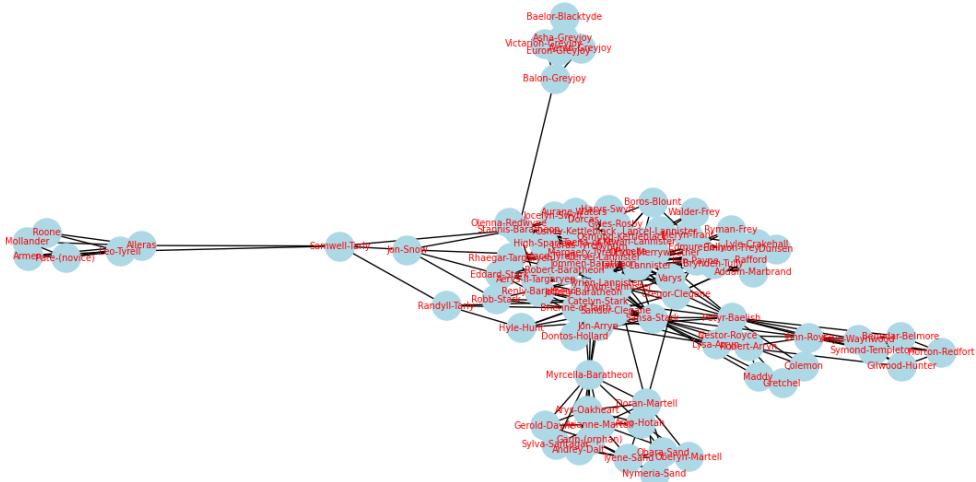
Book 4 Cores Il 4-Core del libro 4 è un po' interessante perché ricompare il Cluster della Night Watch.

Oltre a questo compaiono due nuovi clusters:

- * Il cluster che identifica le vicende della famiglia Martell;
- * Il cluster della famiglia Greyjoy.

```
[ ]: interesting_k_core_4 = nx.k_core(G_4, k=4, core_number=None)
k_core_4_nodes = plot_k_core(interesting_k_core_4, 'K-Core for Book 4', font_size=7)
```

K-Core for Book 4

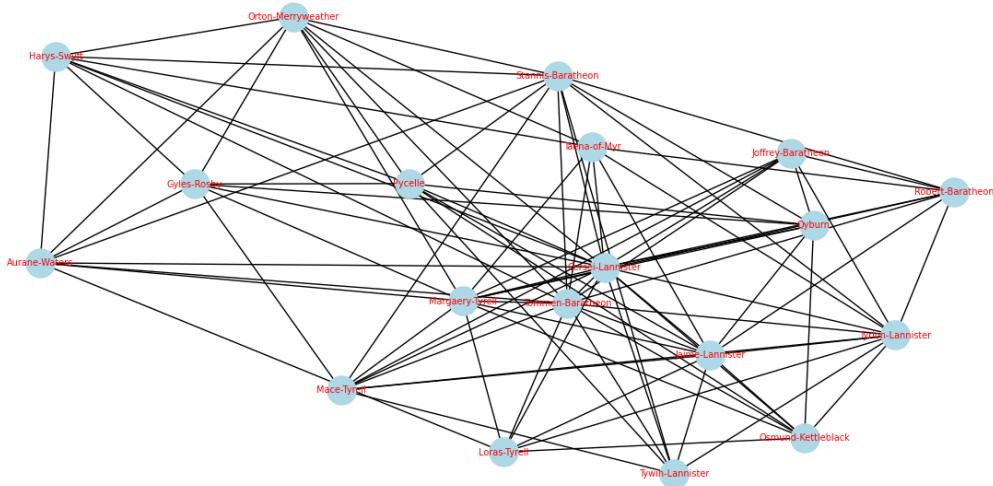


Il core più ristretto del libro 4 è il 7-Core. Questo è difficile da interpretare ma ci ho pensato e sono arrivato a questa spiegazione.

Qui Robert Baratheon è morto e questo Core è come se fosse una fotografia della complessa situazione politica: * Stannis Baratheon sta cercando di rivendicare il trono; * Margaery Tyrell e Loras Tyrell cercano di consolidare il potere della loro famiglia attraverso matrimoni strategici. Margaery sposa Joffrey Baratheon, mentre Loras è il suo fratello; * Qyburn è un ex maestro ed è al servizio dei Lannister; * Harys Swyft, Gyles Rosby, e Orton Merryweather sono nobili che si alleano spesso con i Lannister per averne dei vantaggi vari.

```
[ ]: interesting_k_core_4 = nx.k_core(G_4, k=7, core_number=None)
k_core_4_nodes = plot_k_core(interesting_k_core_4, 'K-Core for Book 4', font_size=7)
```

K-Core for Book 4



Book 5 Cores Dal 5-Core del libro 5 si vedono quasi tutti i cluster delle vicende principali di questo libro.

Da una parte abbiamo tutte le vicende del continente ad est: quindi il cluster di Daenerys Targaryen.

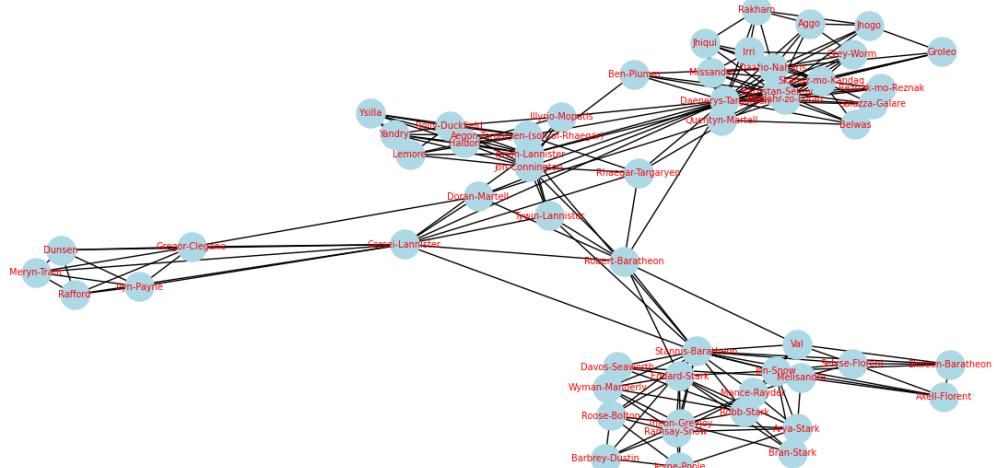
Un piccolo cluster identifica di nuovo Cersei e la guardia reale.

un altro cluster più grande identifica tutte le vicende al nord: comunitativo di Stannis e di Jon-Snow.

L'ultimo cluster più piccolo è quello di Kings Landing.

```
[ ]: interesting_k_core_5 = nx.k_core(G_5, k=5, core_number=None)
k_core_5_nodes = plot_k_core(interesting_k_core_5, 'K-Core for Book 5',
                           font_size=7)
```

K-Core for Book 5



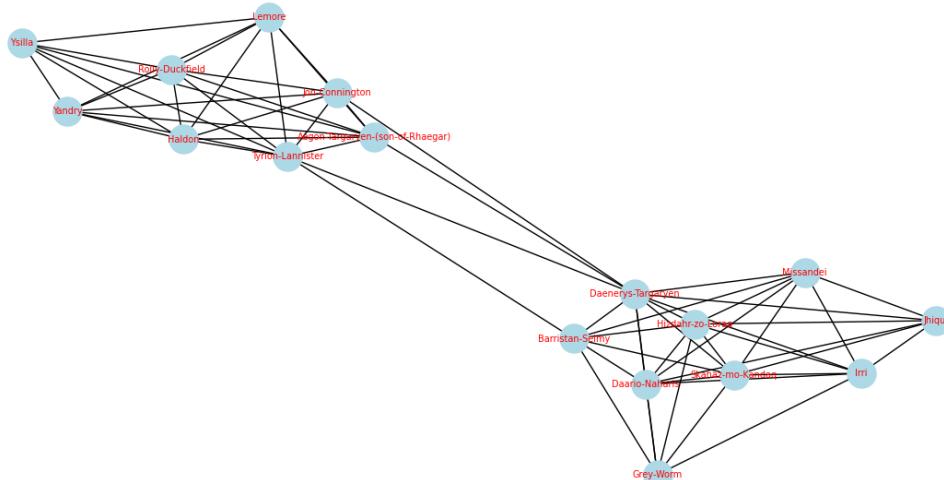
Il 6-Core identifica di fatto la parte principale della trama.

Qui Tyrion ha ucciso suo padre e sta scappando oltre il mare per poi arrivare nel continente di Daenerys.

Aegon Targaryen è quello che qui rischia di mettere i bastoni tra le ruote a Daenerys.

```
[ ]: interesting_k_core_5 = nx.k_core(G_5, k=6, core_number=None)
k_core_5_nodes = plot_k_core(interesting_k_core_5, 'K-Core for Book 5', font_size=7)
```

K-Core for Book 5



K-Components

```
[ ]: k_components_1 = nx.k_components(G_1)
k_components_2 = nx.k_components(G_2)
k_components_3 = nx.k_components(G_3)
k_components_4 = nx.k_components(G_4)
k_components_5 = nx.k_components(G_5)

[ ]: print("\n##### All found K-Components in Book 1 #####")
for key, value in k_components_1.items():
    print(f"\n{key}-Components:")
    for component in value:
        print(f"\t{component}")

print("\n\n##### All found K-Components in Book 2 #####")
for key, value in k_components_2.items():
    print(f"\n{key}-Components:")
    for component in value:
        print(f"\t{component}")

print("\n\n##### All found K-Components in Book 3 #####")
for key, value in k_components_3.items():
    print(f"\n{key}-Components:")
    for component in value:
        print(f"\t{component}")

print("\n\n##### All found K-Components in Book 4 #####")
for key, value in k_components_4.items():
    print(f"\n{key}-Components:")
    for component in value:
        print(f"\t{component}")

print("\n\n##### All found K-Components in Book 5 #####")
for key, value in k_components_5.items():
    print(f"\n{key}-Components:")
    for component in value:
        print(f"\t{component}")
```

All found K-Components in Book 1

11-Components:

```
{'Eddard-Stark', 'Joffrey-Baratheon', 'Arya-Stark', 'Varys', 'Sandor-Clegane', 'Bran-Stark', 'Jory-Cassel', 'Petyr-Baelish', 'Renly-Baratheon', 'Stannis-Baratheon', 'Sansa-Stark', 'Cersei-Lannister', 'Jaime-Lannister', 'Jon-Arryn', 'Robb-Stark', 'Pycelle', 'Tywin-Lannister', 'Catelyn-Stark', 'Jon-Snow', 'Robert-Baratheon', 'Barristan-Selmy', 'Tyrion-Lannister', 'Rodrik-Cassel'}
```

10-Components:

```
{'Eddard-Stark', 'Joffrey-Baratheon', 'Gregor-Clegane', 'Arya-Stark', 'Varys', 'Sandor-Clegane', 'Bran-Stark', 'Loras-Tyrell', 'Jory-Cassel', 'Petyr-Baelish', 'Renly-Baratheon', 'Stannis-Baratheon', 'Sansa-Stark', 'Cersei-Lannister', 'Jaime-Lannister', 'Jon-Arryn', 'Robb-Stark', 'Pycelle', 'Tywin-Lannister', 'Catelyn-Stark', 'Jon-Snow', 'Robert-Baratheon', 'Barristan-Selmy', 'Tyrion-Lannister', 'Rodrik-Cassel'}
```

9-Components:

```
{'Rickon-Stark', 'Joffrey-Baratheon', 'Theon-Greyjoy', 'Varys', 'Sandor-Clegane', 'Bran-Stark', 'Mordane', 'Petyr-Baelish', 'Renly-Baratheon', 'Luwin', 'Sansa-Stark', 'Benjen-Stark', 'Jon-Snow', 'Barristan-Selmy', 'Rodrik-Cassel', 'Eddard-Stark', 'Gregor-Clegane', 'Arya-Stark', 'Loras-Tyrell', 'Jory-Cassel', 'Stannis-Baratheon', 'Cersei-Lannister', 'Jaime-Lannister', 'Jon-Arryn', 'Robb-Stark', 'Tywin-Lannister', 'Pycelle', 'Catelyn-Stark', 'Robert-Baratheon', 'Tyrion-Lannister'}
```

8-Components:

```
{'Rickon-Stark', 'Joffrey-Baratheon', 'Theon-Greyjoy', 'Varys', 'Sandor-Clegane', 'Mordane', 'Bran-Stark', 'Petyr-Baelish', 'Lysa-Arryn', 'Renly-Baratheon', 'Luwin', 'Sansa-Stark', 'Benjen-Stark', 'Jon-Snow', 'Barristan-Selmy', 'Rodrik-Cassel', 'Eddard-Stark', 'Gregor-Clegane', 'Arya-Stark', 'Loras-Tyrell', 'Jory-Cassel', 'Ilyn-Payne', 'Stannis-Baratheon', 'Cersei-Lannister', 'Jaime-Lannister', 'Jon-Arryn', 'Robb-Stark', 'Pycelle', 'Tywin-Lannister', 'Catelyn-Stark', 'Robert-Baratheon', 'Tyrion-Lannister'}
```

7-Components:

```
{'Rickon-Stark', 'Joffrey-Baratheon', 'Janos-Slynt', 'Theon-Greyjoy', 'Varys', 'Sandor-Clegane', 'Bran-Stark', 'Mordane', 'Petyr-Baelish', 'Lysa-Arryn', 'Renly-Baratheon', 'Luwin', 'Sansa-Stark', 'Benjen-Stark', 'Myrcella-Baratheon', 'Edmure-Tully', 'Brynden-Tully', 'Jon-Snow', 'Barristan-Selmy', 'Rodrik-Cassel', 'Tommen-Baratheon', 'Eddard-Stark', 'Gregor-Clegane', 'Arya-Stark', 'Loras-Tyrell', 'Jory-Cassel', 'Ilyn-Payne', 'Stannis-Baratheon', 'Cersei-Lannister', 'Jaime-Lannister', 'Jon-Arryn', 'Robb-Stark', 'Pycelle', 'Tywin-Lannister', 'Catelyn-Stark', 'Robert-Baratheon', 'Tyrion-Lannister'}
```

6-Components:

```
{'Yoren', 'Rickon-Stark', 'Joffrey-Baratheon', 'Janos-Slynt', 'Theon-Greyjoy', 'Varys', 'Sandor-Clegane', 'Bran-Stark', 'Mordane', 'Petyr-Baelish', 'Lysa-Arryn', 'Renly-Baratheon', 'Luwin', 'Sansa-Stark', 'Benjen-Stark', 'Myrcella-Baratheon', 'Edmure-Tully', 'Jeor-Mormont', 'Brynden-Tully', 'Jon-
```

Snow', 'Barristan-Selmy', 'Rodrik-Cassel', 'Tommen-Baratheon', 'Eddard-Stark', 'Gregor-Clegane', 'Arya-Stark', 'Loras-Tyrell', 'Jeyne-Poole', 'Jory-Cassel', 'Ilyn-Payne', 'Stannis-Baratheon', 'Cersei-Lannister', 'Jaime-Lannister', 'Jon-Arryn', 'Robb-Stark', 'Rickard-Karstark', 'Tywin-Lannister', 'Pycelle', 'Catelyn-Stark', 'Robert-Baratheon', 'Tyrion-Lannister'}

5-Components:

{'Beric-Dondarrion', 'Yoren', 'Rickon-Stark', 'Hullen', 'Joffrey-Baratheon', 'Janos-Slynt', 'Theon-Greyjoy', 'Varys', 'Sandor-Clegane', 'Bran-Stark', 'Mordane', 'Petyr-Baelish', 'Lysa-Arryn', 'Renly-Baratheon', 'Luwin', 'Hallis-Mollen', 'Sansa-Stark', 'Benjen-Stark', 'Jon-Umber-(Greatjon)', 'Myrcella-Baratheon', 'Edmure-Tully', 'Walder-Frey', 'Jeor-Mormont', 'Brynden-Tully', 'Vayon-Poole', 'Jon-Snow', 'Barristan-Selmy', 'Rodrik-Cassel', 'Roose-Bolton', 'Tommen-Baratheon', 'Eddard-Stark', 'Gregor-Clegane', 'Meryn-Trant', 'Galbart-Glover', 'Arya-Stark', 'Boros-Blount', 'Loras-Tyrell', 'Jeyne-Poole', 'Jory-Cassel', 'Mycah', 'Ilyn-Payne', 'Stannis-Baratheon', 'Cersei-Lannister', 'Hoster-Tully', 'Jaime-Lannister', 'Rickard-Karstark', 'Robb-Stark', 'Jon-Arryn', 'Tomard', 'Pycelle', 'Tywin-Lannister', 'Catelyn-Stark', 'Robert-Baratheon', 'Tyrion-Lannister'}
{'Marillion', 'Catelyn-Stark', 'Willis-Wode', 'Tyrion-Lannister', 'Bronn', 'Rodrik-Cassel'}
{'Quaro', 'Rakharo', 'Qotho', 'Daenerys-Targaryen', 'Aggo', 'Haggo', 'Jorah-Mormont', 'Jhogo', 'Drogo', 'Mirri-Maz-Duur'}
{'Viserys-Targaryen', 'Daenerys-Targaryen', 'Irri', 'Jorah-Mormont', 'Doreah', 'Drogo', 'Jhiqui', 'Illyrio-Mopatis', 'Robert-Baratheon'}
{'Pypar', 'Rast', 'Jon-Snow', 'Halder', 'Samwell-Tarly', 'Alliser-Thorne'}
{'Jon-Snow', 'Bowen-Marsh', 'Samwell-Tarly', 'Jeor-Mormont', 'Alliser-Thorne', 'Aemon-Targaryen-(Maester-Aemon)'}

4-Components:

{'Beric-Dondarrion', 'Yoren', 'Rickon-Stark', 'Hullen', 'Marq-Piper', 'Theon-Greyjoy', 'Bran-Stark', 'Petyr-Baelish', 'Renly-Baratheon', 'Luwin', 'Hallis-Mollen', 'Sansa-Stark', 'Walder-Frey', 'Benjen-Stark', 'Myrcella-Baratheon', 'Jeor-Mormont', 'Barristan-Selmy', 'Aerys-II-Targaryen', 'Rodrik-Cassel', 'Lyanna-Stark', 'Rhaegar-Targaryen', 'Tommen-Baratheon', 'Eddard-Stark', 'Gregor-Clegane', 'Galbart-Glover', 'Meryn-Trant', 'Cersei-Lannister', 'Alyn', 'Hoster-Tully', 'Jon-Arryn', 'Rickard-Karstark', 'Robb-Stark', 'Tomard', 'Catelyn-Stark', 'Tyrion-Lannister', 'Joffrey-Baratheon', 'Bronn', 'Varys', 'Sandor-Clegane', 'Mordane', 'Willis-Wode', 'Brandon-Stark', 'Lysa-Arryn', 'Jon-Umber-(Greatjon)', 'Edmure-Tully', 'Brynden-Tully', 'Vayon-Poole', 'Jon-Snow', 'Roose-Bolton', 'Daenerys-Targaryen', 'Arya-Stark', 'Boros-Blount', 'Loras-Tyrell', 'Jory-Cassel', 'Jeyne-Poole', 'Mycah', 'Ilyn-Payne', 'Marillion', 'Stannis-Baratheon', 'Jaime-Lannister', 'Pycelle', 'Tywin-Lannister', 'Harwin', 'Robert-Baratheon', 'Janos-Slynt'}
{'Viserys-Targaryen', 'Cohollo', 'Rhaegar-Targaryen', 'Qotho', 'Daenerys-Targaryen', 'Aggo', 'Haggo', 'Jorah-Mormont', 'Jhogo', 'Illyrio-Mopatis', 'Drogo', 'Jhiqui', 'Mirri-Maz-Duur', 'Quaro', 'Rakharo', 'Irri', 'Viserys-Targaryen', 'Cohollo', 'Rhaegar-Targaryen', 'Qotho', 'Daenerys-Targaryen', 'Aggo', 'Haggo', 'Jorah-Mormont', 'Jhogo', 'Illyrio-Mopatis', 'Drogo', 'Jhiqui', 'Mirri-Maz-Duur', 'Quaro', 'Rakharo', 'Irri'}

```

'Doreah', 'Robert-Baratheon'}
{'Grenn', 'Jeor-Mormont', 'Alliser-Thorne', 'Aemon-Targaryen-(Maester-Aemon)', 'Pypar', 'Halder', 'Rast', 'Bowen-Marsh', 'Jon-Snow', 'Samwell-Tarly', 'Todder'}
{'Nan', 'Bran-Stark', 'Luwin', 'Robb-Stark', 'Hodor'}

```

3-Components:

```

'Beric-Dondarrion', 'Marq-Piper', 'Albett', 'Rickon-Stark', 'Hullen',
'Yoren', 'Theon-Greyjoy', 'Robert-Arryn', 'Drogo', 'Bran-Stark', 'Petyr-Baelish',
'Mirri-Maz-Duur', 'Renly-Baratheon', 'Luwin', 'Hallis-Mollen', 'Sansa-Stark',
'Walder-Frey', 'Benjen-Stark', 'Myrcella-Baratheon', 'Grenn', 'Vardis-Egen',
'Jeor-Mormont', 'Irri', 'Jaremy-Rykker', 'Eon-Hunter', 'Barristan-Selmy',
'Aerys-II-Targaryen', 'Rodrik-Cassel', 'Lyanna-Stark', 'Rhaegar-Targaryen',
'Tommen-Baratheon', 'Eddard-Stark', 'Gregor-Clegane', 'Haggo', 'Meryn-Trant',
'Galbart-Glover', 'Aggo', 'Cayn', 'Illyrio-Mopatis', 'Gerold-Hightower',
'Hodor', 'Kevan-Lannister', 'Rakharo', 'Chiggen', 'Alyn', 'Cersei-Lannister',
'Nan', 'Rickard-Karstark', 'Robb-Stark', 'Hoster-Tully', 'Tomard', 'Jon-Arryn',
'Catelyn-Stark', 'Bowen-Marsh', 'Samwell-Tarly', 'Tyrion-Lannister', 'Viserys-Targaryen',
'Chett', 'Cohollo', 'Qotho', 'Joffrey-Baratheon', 'Bronn', 'Halder',
'Varys', 'Jhiqui', 'Sandor-Clegane', 'Willis-Wode', 'Mordane', 'Brandon-Stark',
'Lysa-Arryn', 'Jon-Umber-(Greatjon)', 'Edmure-Tully', 'Aemon-Targaryen-(Maester-Aemon)',
'Brynden-Tully', 'Pypar', 'Vayon-Poole', 'Rast', 'Jon-Snow', 'Osha',
'Roose-Bolton', 'Daenerys-Targaryen', 'Alliser-Thorne', 'Jhogo', 'Arya-Stark',
'Jorah-Mormont', 'Boros-Blount', 'Loras-Tyrell', 'Todder', 'Mycah', 'Jory-Cassel',
'Jeyne-Poole', 'Ilyn-Payne', 'Marillion', 'Quaro', 'Stannis-Baratheon',
'Jaime-Lannister', 'Tywin-Lannister', 'Syrio-Forel', 'Pycelle', 'Harwin',
'Doreah', 'Robert-Baratheon', 'Janos-Slynt'}

```

2-Components:

```

{'Jared-Frey', 'Danwell-Frey', 'Hosteen-Frey'}
{'Morrec', 'Tyrion-Lannister', 'Jyck'}
{'Drogo', 'Ogo', 'Jommo'}
{'Beric-Dondarrion', 'Marq-Piper', 'Albett', 'Rickon-Stark', 'Hullen',
'Robert-Arryn', 'Drogo', 'Daryn-Hornwood', 'Joseth', 'Stiv', 'Luwin', 'Hallis-Mollen',
'Walder-Frey', 'Tytos-Blackwood', 'Jeor-Mormont', 'Irri', 'Raymun-Darry',
'Jaremy-Rykker', 'Nestor-Royce', 'Will-(prologue)', 'Barristan-Selmy',
'Lyanna-Stark', 'Rhaegar-Targaryen', 'Tommen-Baratheon', 'Eddard-Stark',
'Colemon', 'Haggo', 'Meryn-Trant', 'Galbart-Glover', 'Gerold-Hightower', 'Kevan-Lannister',
'Rakharo', 'Alyn', 'Cersei-Lannister', 'Nan', 'Rickard-Karstark',
'Robb-Stark', 'Tomard', 'Bowen-Marsh', 'Viserys-Targaryen', 'Joffrey-Baratheon',
'Bronn', 'Halder', 'Varys', 'Jhiqui', 'Willis-Wode', 'Torrhen-Karstark',
'Edmure-Tully', 'Aemon-Targaryen-(Maester-Aemon)', 'Rickard-Stark', 'Pypar',
'Brynden-Tully', 'Rast', 'Jon-Snow', 'Osha', 'Roose-Bolton', 'Daenerys-Targaryen',
'Alliser-Thorne', 'Jhogo', 'Arya-Stark', 'Stevron-Frey', 'Boros-Blount',
'Karyl-Vance', 'Loras-Tyrell', 'Todder', 'Ilyn-Payne', 'Marillion',
'Balon-Swann', 'Timett', 'Tywin-Lannister', 'Syrio-Forel', 'Pycelle', 'High-Septon-(fat_one)',
'Doreah', 'Wendel-Manderly', 'Yoren', 'Theon-Greyjoy', 'Bran-Stark', 'Petyr-Baelish',
'Mirri-Maz-Duur', 'Renly-Baratheon', 'Shae', 'Sansa'
}
```

Stark', 'Benjen-Stark', 'Myrcella-Baratheon', 'Grenn', 'Vardis-Egen', 'Eon-Hunter', 'Aegon-I-Targaryen', 'Aerys-II-Targaryen', 'Rodrik-Cassel', 'Gregor-Clegane', 'Aggo', 'Cayn', 'Illyrio-Mopatis', 'Shagga', 'Hodor', 'Maege-Mormont', 'Chiggen', 'Hoster-Tully', 'Jon-Arryn', 'Addam-Marbrand', 'Catelyn-Stark', 'Waymar-Royce', 'Samwell-Tarly', 'Tyrion-Lannister', 'Chett', 'Cohollo', 'Heward', 'Qotho', 'Sandor-Clegane', 'Mordane', 'Brandon-Stark', 'Balon-Greyjoy', 'Lysa-Arryn', 'Jon-Umber-(Greatjon)', 'Wyl-(guard)', 'Vayon-Poole', 'Desmond', 'Jorah-Mormont', 'Gared', 'Mycah', 'Jory-Cassel', 'Jeyne-Poole', 'Conn', 'Quaro', 'Stannis-Baratheon', 'Jaime-Lannister', 'Harwin', 'Robert-Baratheon', 'Janos-Slynt'}

1-Components:

{'Marq-Piper', 'Beric-Dondarrion', 'Rickon-Stark', 'Hullen', 'Albett', 'Mace-Tyrell', 'Robert-Arryn', 'Fogo', 'Drogo', 'Ogo', 'Daryn-Hornwood', 'Arthur-Dayne', 'Josef', 'Rhaego', 'Stiv', 'Hallis-Mollen', 'Luwin', 'Gendry', 'Walder-Frey', 'Tyto-Blackwood', 'Jommo', 'Jeor-Mormont', 'Irri', 'Raymun-Darry', 'Jaremy-Rykker', 'Nestor-Royce', 'Wylla', 'Chella', 'Will-(prologue)', 'Barristan-Selmy', 'Lyanna-Stark', 'Rhaegar-Targaryen', 'Danwell-Frey', 'Harys-Swift', 'Tommen-Baratheon', 'Eddard-Stark', 'Colemon', 'Hosteen-Frey', 'Haggo', 'Galbart-Glover', 'Meryn-Trant', 'Mance-Rayder', 'Gerold-Hightower', 'Hugh', 'Dareon', 'Kevan-Lannister', 'Leo-Lefford', 'Rakharo', 'Cersei-Lannister', 'Alyn', 'Nan', 'Rickard-Karstark', 'Robb-Stark', 'Tomard', 'Kurleket', 'Tobho-Mott', 'Morrec', 'Coratt', 'Bowen-Marsh', 'Clydas', 'Viserys-Targaryen', 'Joffrey-Baratheon', 'Bronn', 'Howland-Reed', 'Varys', 'Halder', 'Jhiqui', 'Willis-Wode', 'Lancel-Lannister', 'Torrhen-Karstark', 'Jyck', 'Edmure-Tully', 'Joss', 'Aemon-Targaryen-(Maester-Aemon)', 'Brynden-Tully', 'Varly', 'Rickard-Stark', 'Pypar', 'Jon-Snow', 'Rast', 'Hobb', 'Osha', 'Roose-Bolton', 'Randyll-Tarly', 'Dolf', 'Daenerys-Targaryen', 'Alliser-Thorne', 'Jhogo', 'Arya-Stark', 'Stevron-Frey', 'Boros-Blount', 'Arys-Oakheart', 'Karyl-Vance', 'Loras-Tyrell', 'Todder', 'Ilyn-Payne', 'Marillion', 'Jonos-Bracken', 'Balon-Swann', 'Timett', 'Tywin-Lannister', 'Pycelle', 'Syrio-Forel', 'High-Septon-(fat_one)', 'Doreah', 'Donal-Noye', 'Jafer-Flowers', 'Wendel-Manderly', 'Yoren', 'Theon-Greyjoy', 'Bran-Stark', 'Petyr-Baelish', 'Mirri-Maz-Duur', 'Renly-Baratheon', 'Shae', 'Sansa-Stark', 'Benjen-Stark', 'Myrcella-Baratheon', 'Grenn', 'Vardis-Egen', 'Eon-Hunter', 'Tregar', 'Moreo-Tumitis', 'Aegon-I-Targaryen', 'Maegor-I-Targaryen', 'Dywen', 'Aerys-II-Targaryen', 'Rodrik-Cassel', 'Ulf-son-of-Umar', 'Gregor-Clegane', 'Aggo', 'Porther', 'Cayn', 'Illyrio-Mopatis', 'Mya-Stone', 'Shagga', 'Hodor', 'Maege-Mormont', 'Chiggen', 'Hoster-Tully', 'Othor', 'Jon-Arryn', 'Matthar', 'Addam-Marbrand', 'Catelyn-Stark', 'Waymar-Royce', 'Samwell-Tarly', 'Tyrion-Lannister', 'Oswell-Whent', 'Chett', 'Cohollo', 'Heward', 'Qotho', 'Mord', 'Sandor-Clegane', 'Mordane', 'Brandon-Stark', 'Balon-Greyjoy', 'Lysa-Arryn', 'Paxter-Redwyne', 'Jon-Umber-(Greatjon)', 'Wyl-(guard)', 'Lyn-Corbray', 'Clement-Piper', 'Vayon-Poole', 'Gunthor-son-of-Gurn', 'Desmond', 'Thoros-of-Myr', 'Hali', 'Donnel-Waynwood', 'Jacks', 'Eroeh', 'Jorah-Mormont', 'Gared', 'Mycah', 'Jory-Cassel', 'Mychel-Redfort', 'Jeyne-Poole', 'Conn', 'Quaro', 'Stannis-Baratheon', 'Masha-Heddle', 'Jaime-Lannister', 'Harwin', 'Robert-Baratheon', 'Jared-Frey', 'Janos-Slynt'}

All found K-Components in Book 2

11-Components:

```
{'Amory-Lorch', 'Cersei-Lannister', 'Polliver', 'Chiswyck', 'Gregor-Clegane', 'Joffrey-Baratheon', 'Meryn-Trant', 'Rafford', 'Dunsen', 'Sandor-Clegane', 'Ilyn-Payne', 'Tickler'}
```

10-Components:

```
{'Amory-Lorch', 'Cersei-Lannister', 'Polliver', 'Chiswyck', 'Gregor-Clegane', 'Joffrey-Baratheon', 'Meryn-Trant', 'Rafford', 'Dunsen', 'Sandor-Clegane', 'Ilyn-Payne', 'Tickler'}
```

9-Components:

```
{'Stannis-Baratheon', 'Sansa-Stark', 'Cersei-Lannister', 'Jaime-Lannister', 'Robb-Stark', 'Eddard-Stark', 'Joffrey-Baratheon', 'Tywin-Lannister', 'Arya-Stark', 'Varys', 'Robert-Baratheon', 'Petyr-Baelish', 'Tyrion-Lannister', 'Renly-Baratheon'}
```

```
{'Amory-Lorch', 'Cersei-Lannister', 'Polliver', 'Chiswyck', 'Gregor-Clegane', 'Joffrey-Baratheon', 'Meryn-Trant', 'Rafford', 'Dunsen', 'Sandor-Clegane', 'Ilyn-Payne', 'Tickler'}
```

8-Components:

```
{'Amory-Lorch', 'Cersei-Lannister', 'Polliver', 'Chiswyck', 'Ilyn-Payne', 'Gregor-Clegane', 'Joffrey-Baratheon', 'Meryn-Trant', 'Rafford', 'Arya-Stark', 'Sandor-Clegane', 'Dunsen', 'Tickler'}
```

```
{'Stannis-Baratheon', 'Sansa-Stark', 'Cersei-Lannister', 'Jaime-Lannister', 'Robb-Stark', 'Eddard-Stark', 'Joffrey-Baratheon', 'Tywin-Lannister', 'Arya-Stark', 'Varys', 'Robert-Baratheon', 'Petyr-Baelish', 'Tyrion-Lannister', 'Renly-Baratheon'}
```

7-Components:

```
{'Amory-Lorch', 'Chiswyck', 'Joffrey-Baratheon', 'Bronn', 'Varys', 'Sandor-Clegane', 'Lancel-Lannister', 'Petyr-Baelish', 'Renly-Baratheon', 'Sansa-Stark', 'Myrcella-Baratheon', 'Jacelyn-Bywater', 'Gyles-Rosby', 'Polliver', 'Dunsen', 'Tommen-Baratheon', 'Eddard-Stark', 'Gregor-Clegane', 'Meryn-Trant', 'Rafford', 'Arya-Stark', 'Boros-Blount', 'Tyrion-Lannister', 'Ilyn-Payne', 'Stannis-Baratheon', 'Cersei-Lannister', 'Jaime-Lannister', 'Robb-Stark', 'Pycelle', 'Tywin-Lannister', 'Robert-Baratheon', 'Janos-Slynt', 'Tickler'}
```

6-Components:

```
{'Amory-Lorch', 'Cleos-Frey', 'Rickon-Stark', 'Chiswyck', 'Joffrey-Baratheon', 'Bronn', 'Janos-Slynt', 'Varys', 'Sandor-Clegane', 'Bran-Stark', 'Lancel-Lannister', 'Petyr-Baelish', 'Renly-Baratheon', 'Sansa-Stark', 'Myrcella-Baratheon', 'Jacelyn-Bywater', 'Jon-Snow', 'Gyles-Rosby', 'Polliver', 'Dunsen', 'Tommen-Baratheon', 'Weese', 'Eddard-Stark', 'Gregor-Clegane', 'Meryn-Trant', 'Rafford', 'Arya-Stark', 'Boros-Blount', 'Loras-Tyrell', 'Ilyn-Payne',
```

'Mandon-Moore', 'Stannis-Baratheon', 'Cersei-Lannister', 'Jaime-Lannister',
'Robb-Stark', 'Pycelle', 'Tywin-Lannister', 'Catelyn-Stark', 'Robert-Baratheon',
'Tyrion-Lannister', 'Tickler'}
{'Hodor', 'Rickon-Stark', 'Meera-Reed', 'Jojen-Reed', 'Bran-Stark',
'Luwin', 'Osha'}

5-Components:

{'Amory-Lorch', 'Cleos-Frey', 'Rickon-Stark', 'Chiswyck', 'Joffrey-Baratheon', 'Bronn', 'Janos-Slynt', 'Theon-Greyjoy', 'Varys', 'Sandor-Clegane', 'Bran-Stark', 'Lancel-Lannister', 'Balon-Greyjoy', 'Petyr-Baelish', 'Renly-Baratheon', 'Luwin', 'Sansa-Stark', 'Myrcella-Baratheon', 'Jacelyn-Bywater', 'Jon-Snow', 'Tanda-Stokeworth', 'Gyles-Rosby', 'Rodrik-Cassel', 'Dunsen', 'Polliver', 'Osha', 'Roose-Bolton', 'Meera-Reed', 'Tommen-Baratheon', 'Weese', 'Eddard-Stark', 'Gregor-Clegane', 'Meryn-Trant', 'Rafford', 'Arya-Stark', 'Boros-Blount', 'Ramsay-Snow', 'Brienne-of-Tarth', 'Loras-Tyrell', 'Osmund-Kettleblack', 'Ilyn-Payne', 'Hodor', 'Mandon-Moore', 'Stannis-Baratheon', 'Cersei-Lannister', 'Jaime-Lannister', 'Margaery-Tyrell', 'Jojen-Reed', 'Robb-Stark', 'Balon-Swann', 'Pycelle', 'Tywin-Lannister', 'Catelyn-Stark', 'Robert-Baratheon', 'Tyrion-Lannister', 'Tickler'}

{'Yoren', 'Woth', 'Hot-Pie', 'Lommy-Greenhands', 'Gendry', 'Arya-Stark'}

4-Components:

{'Amory-Lorch', 'Yoren', 'Cleos-Frey', 'Rickon-Stark', 'Mace-Tyrell', 'Stafford-Lannister', 'Theon-Greyjoy', 'Rorge', 'Bran-Stark', 'Petyr-Baelish', 'Renly-Baratheon', 'Luwin', 'Gendry', 'Biter', 'Sansa-Stark', 'Myrcella-Baratheon', 'Mathis-Rowan', 'Tanda-Stokeworth', 'Barristan-Selmy', 'Gyles-Rosby', 'Aerys-II-Targaryen', 'Rodrik-Cassel', 'Polliver', 'Dunsen', 'Dontos-Hollard', 'Meera-Reed', 'Tommen-Baratheon', 'Eddard-Stark', 'Gregor-Clegane', 'Meryn-Trant', 'Brienne-of-Tarth', 'Osmund-Kettleblack', 'Hodor', 'Cersei-Lannister', 'Walder-Frey-(son-of-Merrett)', 'Margaery-Tyrell', 'Jojen-Reed', 'Robb-Stark', 'Catelyn-Stark', 'Tyrion-Lannister', 'Jaqen-Hghar', 'Chiswyck', 'Joffrey-Baratheon', 'Bronn', 'Varys', 'Sandor-Clegane', 'Lancel-Lannister', 'Balon-Greyjoy', 'Jacelyn-Bywater', 'Jon-Snow', 'Randyll-Tarly', 'Roose-Bolton', 'Osha', 'Weese', 'Donella-Hornwood', 'Rafford', 'Arya-Stark', 'Woth', 'Hot-Pie', 'Boros-Blount', 'Ramsay-Snow', 'Loras-Tyrell', 'Lommy-Greenhands', 'Ilyn-Payne', 'Mandon-Moore', 'Stannis-Baratheon', 'Jaime-Lannister', 'Balon-Swann', 'Tywin-Lannister', 'Pycelle', 'Wyman-Manderly', 'Weasel', 'Robert-Baratheon', 'Vargo-Hoat', 'Janos-Slynt', 'Tickler'}
{'Belwas', 'Jorah-Mormont', 'Daenerys-Targaryen', 'Aggo', 'Jhogo'}
{'Alester-Florent', 'Stannis-Baratheon', 'Davos-Seaworth', 'Cortnay-Penrose', 'Renly-Baratheon'}
{'Mance-Rayder', 'Craster', 'Jon-Snow', 'Jeor-Mormont', 'Qhorin-Halfhand', 'Thoren-Smallwood'}

3-Components:

{'Amory-Lorch', 'Yoren', 'Cleos-Frey', 'Rickon-Stark', 'Harren-Hoare', 'Mace-Tyrell', 'Asha-Greyjoy', 'Stafford-Lannister', 'Theon-Greyjoy', 'Rorge', 'Mikken', 'Belwas', 'Drogo', 'Bran-Stark', 'Beth-Cassel', 'Alebelly', 'Petyr-

Baelish', 'Renly-Baratheon', 'Luwin', 'Gendry', 'Shae', 'Biter', 'Sansa-Stark',
 'Selyse-Florent', 'Myrcella-Baratheon', 'Mathis-Rowan', 'Jeor-Mormont', 'Emmon-
 Cuy', 'Aegon-I-Targaryen', 'Chella', 'Tanda-Stokeworth', 'Barristan-Selmy',
 'Gyles-Rosby', 'Pono', 'Aerys-II-Targaryen', 'Rodrik-Cassel', 'Dunsen',
 'Polliver', 'Dontos-Hollard', 'Rhaegar-Targaryen', 'Meera-Reed', 'Dagmer',
 'Tommen-Baratheon', 'Eddard-Stark', 'Gregor-Clegane', 'Aggo', 'Meryn-Trant',
 'Wex-Pyke', 'Brienne-of-Tarth', 'Cressen', 'Shagga', 'Osmund-Kettleblack',
 'Hodor', 'Rakharo', 'Cersei-Lannister', 'Nan', 'Margaery-Tyrell', 'Jojen-Reed',
 'Robb-Stark', 'Walder-Frey-(son-of-Merrett)', 'Hoster-Tully', 'Catelyn-Stark',
 'Davos-Seaworth', 'Doran-Martell', 'Melisandre', 'Tyrion-Lannister', 'Viserys-
 Targaryen', 'Devan-Seaworth', 'Jaqen-Hghar', 'Bryce-Caron', 'Chiswyck',
 'Joffrey-Baratheon', 'Bronn', 'Varys', 'Sandor-Clegane', 'Lancel-Lannister',
 'Balon-Greyjoy', 'Lysa-Arryn', 'Tysha', 'Edmure-Tully', 'Cley-Cerwyn', 'Vyman',
 'Jacelyn-Bywater', 'Brynden-Tully', 'Jon-Snow', 'Podrick-Payne', 'Osha', 'Roose-
 Bolton', 'Randyll-Tarly', 'Weese', 'Daenerys-Targaryen', 'Robar-Royce',
 'Donella-Hornwood', 'Alliser-Thorne', 'Rafford', 'Arya-Stark', 'Jhogo', 'Woth',
 'Boros-Blount', 'Jorah-Mormont', 'Arys-Oakheart', 'Hot-Pie', 'Ramsay-Snow',
 'Cortnay-Penrose', 'Loras-Tyrell', 'Lommy-Greenhands', 'Ilyn-Payne', 'Mandon-
 Moore', 'Stannis-Baratheon', 'Alester-Florent', 'Jaime-Lannister', 'Balon-
 Swann', 'Timett', 'Tywin-Lannister', 'Pycelle', 'Wyman-Manderly', 'Weasel',
 'Robert-Baratheon', 'Vargo-Hoat', 'Janos-Slynt', 'Tickler'}
 {'Grenn', 'Eddison-Tollett', 'Jeor-Mormont', 'Qhorin-Halfhand', 'Aemon-
 Targaryen-(Maester-Aemon)', 'Dywen', 'Mance-Rayder', 'Ottyn-Wythers', 'Craster',
 'Jon-Snow', 'Samwell-Tarly', 'Jarman-Buckwell', 'Thoren-Smallwood'}
 {'Stonesnake', 'Jon-Snow', 'Dalbridge', 'Rattleshirt', 'Qhorin-
 Halfhand', 'Ygritte', 'Ebbens'}

2-Components:

{'Amabel', 'Harra', 'Arya-Stark'}
 {'Daenerys-Targaryen', 'Irri', 'Jhiqui'}
 {'Daenerys-Targaryen', 'Xaro-Xhoan-Daxos', 'Pyat-Pree'}
 {'Amory-Lorch', 'Rickon-Stark', 'Mace-Tyrell', 'Robett-Glover',
 'Mikken', 'Belwas', 'Rorge', 'Drogo', 'Koss', 'Beth-Cassel', 'Luwin', 'Gendry',
 'Hallis-Mollen', 'Urzen', 'Walder-Frey', 'Mathis-Rowan', 'Jeor-Mormont',
 'Jaremy-Rykker', 'Emmon-Cuy', 'Ottyn-Wythers', 'Chella', 'Tanda-Stokeworth',
 'Barristan-Selmy', 'Gyles-Rosby', 'Patchface', 'Dunsen', 'Dagmer', 'Rhaegar-
 Targaryen', 'Meera-Reed', 'Dontos-Hollard', 'Stonesnake', 'Tommen-Baratheon',
 'Eddard-Stark', 'Meryn-Trant', 'Aggar', 'Mance-Rayder', 'Brienne-of-Tarth',
 'Cressen', 'Osmund-Kettleblack', 'Rakharo', 'Cersei-Lannister', 'Nan', 'Jojen-
 Reed', 'Leobald-Tallhart', 'Robb-Stark', 'Walder-Frey-(son-of-Merrett)',
 'Margaery-Tyrell', 'Davos-Seaworth', 'Doran-Martell', 'Eddison-Tollett',
 'Jarman-Buckwell', 'Werlag', 'Viserys-Targaryen', 'Bryce-Caron', 'Joffrey-
 Baratheon', 'Bronn', 'Ygritte', 'Ballabar', 'Benfred-Tallhart', 'Varys',
 'Lancel-Lannister', 'Edmure-Tully', 'Shireen-Baratheon', 'Aemon-
 Targaryen-(Maester-Aemon)', 'Vyman', 'Jacelyn-Bywater', 'Brynden-Tully', 'Jon-
 Snow', 'Monford-Velaryon', 'Allard-Seaworth', 'Podrick-Payne', 'Thoren-
 Smallwood', 'Randyll-Tarly', 'Roose-Bolton', 'Osha', 'Weese', 'Daenerys-
 Targaryen', 'Donella-Hornwood', 'Robar-Royce', 'Alliser-Thorne', 'Reek',

'Jhogo', 'Arya-Stark', 'Rafford', 'Woth', 'Boros-Blount', 'Desmond-Grell',
 'Arys-Oakheart', 'Loras-Tyrell', 'Ilyn-Payne', 'Utherydes-Wayn', 'Balon-Swann',
 'Timett', 'Tywin-Lannister', 'Syrio-Forel', 'Pycelle', 'Weasel', 'Osney-
 Kettleblack', 'Wendel-Manderly', 'Tickler', 'Yoren', 'Cleos-Frey', 'Harren-
 Hoare', 'Asha-Greyjoy', 'Stafford-Lannister', 'Theon-Greyjoy', 'Pylos',
 'Steffon-Baratheon', 'Craster', 'Alebelly', 'Bran-Stark', 'Petyr-Baelish',
 'Renly-Baratheon', 'Shae', 'Biter', 'Sansa-Stark', 'Selyse-Florent', 'Myrcella-
 Baratheon', 'Grenn', 'Squint', 'Aegon-I-Targaryen', 'Dywen', 'Salladhor-Saan',
 'Axell-Florent', 'Pono', 'Aerys-II-Targaryen', 'Rodrik-Cassel', 'Polliver',
 'Hallyne', 'Dalbridge', 'Gregor-Clegane', 'Aggo', 'Wex-Pyke', 'Victarion-
 Greyjoy', 'Lollys-Stokeworth', 'Shagga', 'Hodor', 'Hoster-Tully', 'Garlan-
 Tyrell', 'Preston-Greenfield', 'Catelyn-Stark', 'Vylarr', 'Samwell-Tarly',
 'Rattleshirt', 'Melisandre', 'Tyrion-Lannister', 'Walder-Frey-(son-of-Jammos)',
 'Devan-Seaworth', 'Jaen-Hghar', 'Colen-of-Greenpools', 'Chiswyck', 'Allar-
 Deem', 'Qhorin-Halfhand', 'Ebben', 'Sandor-Clegane', 'Lorren', 'Balon-Greyjoy',
 'Lysa-Arryn', 'Tysha', 'Cley-Cerwyn', 'Chayle', 'Guyard-Morrigen', 'Gynir',
 'Jorah-Mormont', 'Hot-Pie', 'Ramsay-Snow', 'Cortnay-Penrose', 'Drennan', 'Lommy-
 Greenhands', 'Hayhead', 'Perwyn-Frey', 'Alester-Florent', 'Stannis-Baratheon',
 'Mandon-Moore', 'Jaime-Lannister', 'Stygg', 'Wyman-Manderly', 'Aeron-Greyjoy',
 'Robert-Baratheon', 'Palla', 'Vargo-Hoat', 'Janos-Slynt'}
 {'Rhaenys-Targaryen', 'Visenya-Targaryen', 'Aegon-I-Targaryen'}

1-Components:

{'Amory-Lorch', 'Beric-Dondarrion', 'Harra', 'Steffon-Varner', 'Robert-
 Arryn', 'Rorge', 'Mikken', 'Mebble', 'Moon-Boy', 'Drogo', 'Beth-Cassel', 'Jon-
 Fossoway', 'Hallis-Mollen', 'Gendry', 'Urzen', 'Walder-Frey', 'Mathis-Rowan',
 'Jaremy-Rykker', 'Ottyn-Wythers', 'Tanda-Stokeworth', 'Farlen', 'Dancy',
 'Dunsen', 'Dontos-Hollard', 'Stonesnake', 'Meera-Reed', 'Rhaegar-Targaryen',
 'Bael-the-Bard', 'Meryn-Trant', 'Mance-Rayder', 'Rhaenys-Targaryen', 'Trystane-
 Martell', 'Osmund-Kettleblack', 'Rakharo', 'Walder-Frey-(son-of-Merrett)',
 'Jojen-Reed', 'Rickard-Karstark', 'Leobald-Tallhart', 'Alayaya', 'Qyburn',
 'Davos-Seaworth', 'Daeron-II-Targaryen', 'Doran-Martell', 'Eddison-Tollett',
 'Werlag', 'Gariss', 'Joffrey-Baratheon', 'Bronn', 'Ygritte', 'Benfred-Tallhart',
 'Howland-Reed', 'Jhiqui', 'Edmure-Tully', 'Shireen-Baratheon', 'Aemon-
 Targaryen-(Maester-Aemon)', 'Jon-Snow', 'Monford-Velaryon', 'Podrick-Payne',
 'Thoren-Smallwood', 'Aerys-I-Targaryen', 'Jason-Mallister', 'Dolf', 'Robar-
 Royce', 'Daenerys-Targaryen', 'Reek', 'Desmond-Grell', 'Boros-Blount',
 'Jonquil', 'Arys-Oakheart', 'Loras-Tyrell', 'Tywin-Lannister', 'Syrio-Forel',
 'Weasel', 'High-Septon-(fat_one)', 'Doreah', 'Dobber', 'Osney-Kettleblack',
 'Wendel-Manderly', 'Hobber-Redwyne', 'Tickler', 'Yoren', 'Cleos-Frey', 'Harren-
 Hoare', 'Asha-Greyjoy', 'Stafford-Lannister', 'Shadd', 'Steffon-Baratheon',
 'Craster', 'Alebelly', 'Mirri-Maz-Duur', 'Xaro-Xhoan-Daxos', 'Shae', 'Sansa-
 Stark', 'Maron-Greyjoy', 'Florian-the-Fool', 'Dywen', 'Axell-Florent', 'Pono',
 'Aerys-II-Targaryen', 'Rodrik-Cassel', 'Polliver', 'Hallyne', 'Maekar-I-
 Targaryen', 'Gregor-Clegane', 'Wex-Pyke', 'Lollys-Stokeworth', 'Hoster-Tully',
 'Garlan-Tyrell', 'Osfryd-Kettleblack', 'Catelyn-Stark', 'Rattleshirt', 'Eldon-
 Estermont', 'Tyrion-Lannister', 'Qhorin-Halfhand', 'Ebben', 'Amabel', 'Sandor-
 Clegane', 'Lorren', 'Lysa-Arryn', 'Tysha', 'Cley-Cerwyn', 'Guyard-Morrigen',

'Gynir', 'Thoros-of-Myr', 'Jorah-Mormont', 'Cortnay-Penrose', 'Lommy-Greenhands', 'Perwyn-Frey', 'Stannis-Baratheon', 'Jaime-Lannister', 'Stygg', 'Robert-Baratheon', 'Palla', 'Aeron-Greyjoy', 'Vargo-Hoat', 'Praed', 'Lark', 'Olyvar-Frey', 'Rickon-Stark', 'Mace-Tyrell', 'Kyra', 'Robett-Glover', 'Belwas', 'Koss', 'Luwin', 'Kurz', 'Jeor-Mormont', 'Irri', 'Quaith', 'Emmon-Cuy', 'Chella', 'Barristan-Selmy', 'Gyles-Rosby', 'Patchface', 'Lynesse-Hightower', 'Dagmer', 'Tommen-Baratheon', 'Murch', 'Eddard-Stark', 'Aggar', 'Aegon-V-Targaryen', 'Brienne-of-Tarth', 'Cressen', 'Cersei-Lannister', 'Nan', 'Margaery-Tyrell', 'Robb-Stark', 'Jarman-Buckwell', 'Viserys-Targaryen', 'Bryce-Caron', 'Hake', 'Ballabar', 'Horas-Redwyne', 'Varys', 'Lancel-Lannister', 'Matthos-Seaworth', 'Arwyn-Oakheart', 'Visenya-Targaryen', 'Vyman', 'Jacelyn-Bywater', 'Brynden-Tully', 'Euron-Greyjoy', 'Allard-Seaworth', 'Osha', 'Roose-Bolton', 'Randyll-Tarly', 'Mors-Umber', 'Helman-Tallhart', 'Weese', 'Donella-Hornwood', 'Colen-of-Greenpools', 'Halys-Hornwood', 'Alliser-Thorne', 'Rafford', 'Arya-Stark', 'Jhogo', 'Woth', 'Ilyn-Payne', 'Imry-Florent', 'Utherydes-Wayn', 'Balon-Swann', 'Timett', 'Pycelle', 'Captain-Myraham-Daughter', 'Marya-Seaworth', 'Captain-Myraham', 'Selwyn-Tarth', 'Theon-Greyjoy', 'Pylos', 'Bran-Stark', 'Martyn-Rivers', 'Petyr-Baelish', 'Renly-Baratheon', 'Biter', 'Selyse-Florent', 'Benjen-Stark', 'Myrcella-Baratheon', 'Grenn', 'Chataya', 'Squint', 'Aegon-I-Targaryen', 'Salladhor-Saan', 'Tarber', 'Dalbridge', 'Aggo', 'Victarion-Greyjoy', 'Shagga', 'Hodor', 'Pyat-Pree', 'Preston-Greenfield', 'Vylarr', 'Samwell-Tarly', 'Melisandre', 'Walder-Frey-(son-of-Jammos)', 'Chett', 'Devan-Seaworth', 'Jaqen-Hghar', 'Chiswyck', 'Allar-Deem', 'Ardrian-Celtigar', 'Balon-Greyjoy', 'Paxter-Redwyne', 'Jon-Umber-(Greatjon)', 'Aenys-Frey', 'Chayle', 'Aron-Santagar', 'High-Septon-(Tyrions)', 'Lorent-Caswell', 'Hot-Pie', 'Ramsay-Snow', 'Drennan', 'Hayhead', 'Gelmarr', 'Mandon-Moore', 'Alester-Florent', 'Cutjack', 'Wyman-Manderly', 'Poxy-Tym', 'Janos-Slynt'}

All found K-Components in Book 3

10-Components:

{'Cersei-Lannister', 'Polliver', 'Ilyn-Payne', 'Gregor-Clegane', 'Joffrey-Baratheon', 'Meryn-Trant', 'Rafford', 'Arya-Stark', 'Sandor-Clegane', 'Dunsen', 'Tickler'}

9-Components:

{'Cersei-Lannister', 'Polliver', 'Ilyn-Payne', 'Gregor-Clegane', 'Joffrey-Baratheon', 'Meryn-Trant', 'Rafford', 'Arya-Stark', 'Sandor-Clegane', 'Dunsen', 'Tickler'}

8-Components:

{'Polliver', 'Tommen-Baratheon', 'Joffrey-Baratheon', 'Bronn', 'Gregor-Clegane', 'Meryn-Trant', 'Rafford', 'Arya-Stark', 'Varys', 'Sandor-Clegane', 'Oberyn-Martell', 'Ilyn-Payne', 'Stannis-Baratheon', 'Sansa-Stark', 'Tickler', 'Cersei-Lannister', 'Jaime-Lannister', 'Robb-Stark', 'Tywin-Lannister', 'Catelyn-Stark', 'Robert-Baratheon', 'Tyrion-Lannister', 'Dunsen'}

7-Components:

{'Roose-Bolton', 'Tommen-Baratheon', 'Joffrey-Baratheon', 'Bronn', 'Gregor-Clegane', 'Meryn-Trant', 'Rafford', 'Arya-Stark', 'Varys', 'Sandor-Clegane', 'Brienne-of-Tarth', 'Loras-Tyrell', 'Osmund-Kettleblack', 'Oberyn-Martell', 'Renly-Baratheon', 'Ilyn-Payne', 'Kevan-Lannister', 'Stannis-Baratheon', 'Sansa-Stark', 'Tickler', 'Cersei-Lannister', 'Jaime-Lannister', 'Margaery-Tyrell', 'Robb-Stark', 'Tywin-Lannister', 'Catelyn-Stark', 'Robert-Baratheon', 'Elmar-Frey', 'Tyrion-Lannister', 'Dunsen', 'Polliver'}

6-Components:

{'Beric-Dondarrion', 'Willas-Tyrell', 'Mace-Tyrell', 'Greenbeard', 'Bran-Stark', 'Renly-Baratheon', 'Gendry', 'Sansa-Stark', 'Walder-Frey', 'Olenna-Redwyne', 'Elmar-Frey', 'Polliver', 'Dunsen', 'Tommen-Baratheon', 'Gregor-Clegane', 'Meryn-Trant', 'Brienne-of-Tarth', 'Osmund-Kettleblack', 'Oberyn-Martell', 'Kevan-Lannister', 'Lothar-Frey', 'Hoster-Tully', 'Cersei-Lannister', 'Margaery-Tyrell', 'Robb-Stark', 'Addam-Marbrand', 'Catelyn-Stark', 'Lem', 'Tyrion-Lannister', 'Joffrey-Baratheon', 'Bronn', 'Varys', 'Sandor-Clegane', 'Lysa-Arryn', 'Anguy', 'Edmure-Tully', 'Brynden-Tully', 'Tom-of-Sevenstreams', 'Jon-Snow', 'Thoros-of-Myr', 'Roose-Bolton', 'Rafford', 'Arya-Stark', 'Boros-Blount', 'Loras-Tyrell', 'Tansy', 'Ilyn-Payne', 'Stannis-Baratheon', 'Jaime-Lannister', 'Balon-Swann', 'Tywin-Lannister', 'Pycelle', 'Harwin', 'Robert-Baratheon', 'Tickler'}

{'Alliser-Thorne', 'Qhorin-Halfhand', 'Mance-Rayder', 'Craster', 'Denys-Mallister', 'Stannis-Baratheon', 'Grenn', 'Jeor-Mormont', 'Aemon-Targaryen-(Maester-Aemon)', 'Samwell-Tarly', 'Pypar', 'Jon-Snow', 'Bowen-Marsh', 'Donal-Noye', 'Eddison-Tollett', 'Cotter-Pyke', 'Janos-Slynt', 'Othell-Yarwyck'}

{'Jarl', 'Harma', 'Tormund', 'Ygritte', 'Mance-Rayder', 'Styr', 'Jon-Snow', 'Rattleshirt'}

5-Components:

{'Beric-Dondarrion', 'Willas-Tyrell', 'Mace-Tyrell', 'Greenbeard', 'Bran-Stark', 'Renly-Baratheon', 'Gendry', 'Sansa-Stark', 'Walder-Frey', 'Olenna-Redwyne', 'Elmar-Frey', 'Dunsen', 'Polliver', 'Tommen-Baratheon', 'Gregor-Clegane', 'Meryn-Trant', 'Brienne-of-Tarth', 'Osmund-Kettleblack', 'Oberyn-Martell', 'Kevan-Lannister', 'Lothar-Frey', 'Cersei-Lannister', 'Garlan-Tyrell', 'Margaery-Tyrell', 'Hoster-Tully', 'Robb-Stark', 'Rickard-Karstark', 'Addam-Marbrand', 'Catelyn-Stark', 'Lem', 'Tyrion-Lannister', 'Joffrey-Baratheon', 'Bronn', 'Varys', 'Sandor-Clegane', 'Lysa-Arryn', 'Roslin-Frey', 'Anguy', 'Jon-Umber-(Greatjon)', 'Edmure-Tully', 'Brynden-Tully', 'Tom-of-Sevenstreams', 'Jon-Snow', 'Thoros-of-Myr', 'Roose-Bolton', 'Elia-Martell', 'Rafford', 'Arya-Stark', 'Boros-Blount', 'Loras-Tyrell', 'Tansy', 'Ilyn-Payne', 'Stannis-Baratheon', 'Jaime-Lannister', 'Balon-Swann', 'Pycelle', 'Tywin-Lannister', 'Harwin', 'Robert-Baratheon', 'Tickler'}

{'Jon-Snow', 'Satin', 'Spare-Boot', 'Hareth-(Moles-Town)', 'Owen', 'Mully', 'Kegs'}

{'Qhorin-Halfhand', 'Alliser-Thorne', 'Mance-Rayder', 'Craster', 'Denys-Mallister', 'Bedwyck', 'Stannis-Baratheon', 'Grenn', 'Eddison-Tollett', 'Jeor-Mormont', 'Aemon-Targaryen-(Maester-Aemon)', 'Dywen', 'Pypar', 'Jon-Snow',

'Bowen-Marsh', 'Donal-Noye', 'Samwell-Tarly', 'Hobb', 'Cotter-Pyke', 'Janos-Slynt', 'Othell-Yarwyck'}
{'Jarl', 'Harma', 'Tormund', 'Ygritte', 'Mance-Rayder', 'Styr', 'Jon-Snow', 'Rattleshirt'}

4-Components:

{'Amory-Lorch', 'Beric-Dondarrion', 'Willas-Tyrell', 'Lark', 'Marq-Piper', 'Mace-Tyrell', 'Mully', 'Greenbeard', 'Pylos', 'Gilly', 'Craster', 'Bran-Stark', 'Renly-Baratheon', 'Gendry', 'Shae', 'Sansa-Stark', 'Walder-Frey', 'Selyse-Florent', 'Benjen-Stark', 'Myrcella-Baratheon', 'Grenn', 'Spare-Boot', 'Jeor-Mormont', 'Dywen', 'Olenna-Redwyne', 'Elmar-Frey', 'Tanda-Stokeworth', 'Salladhor-Saan', 'Hareth-(Moles-Town)', 'Axell-Florent', 'Kegs', 'Dunsen', 'Polliver', 'Dontos-Hollard', 'Meera-Reed', 'Tommen-Baratheon', 'Gregor-Clegane', 'Meryn-Trant', 'Tormund', 'Mance-Rayder', 'Lollys-Stokeworth', 'Brienne-of-Tarth', 'Ryk', 'Osmund-Kettleblack', 'Oberyn-Martell', 'Small-Paul', 'Kevan-Lannister', 'Lothar-Frey', 'Cersei-Lannister', 'Hoster-Tully', 'Margaery-Tyrell', 'Rickard-Karstark', 'Jojen-Reed', 'Robb-Stark', 'Garlan-Tyrell', 'Addam-Marbrand', 'Catelyn-Stark', 'Davos-Seaworth', 'Bowen-Marsh', 'Clydas', 'Eddison-Tollett', 'Samwell-Tarly', 'Rattleshirt', 'Lem', 'Melisandre', 'Tyrion-Lannister', 'Chett', 'Joffrey-Baratheon', 'Bronn', 'Ygritte', 'Qhorin-Halfhand', 'Varys', 'Sandor-Clegane', 'Denys-Mallister', 'Bedwyck', 'Satin', 'Lysa-Arryn', 'Roslin-Frey', 'Anguy', 'Jarl', 'Jon-Umber-(Greatjon)', 'Edmure-Tully', 'Shireen-Baratheon', 'Aemon-Targaryen-(Maester-Aemon)', 'Brynden-Tully', 'Pypar', 'Tom-of-Sevenstreams', 'Jon-Snow', 'Hobb', 'Podrick-Payne', 'Owen', 'Cotter-Pyke', 'Thoros-of-Myr', 'Othell-Yarwyck', 'Roose-Bolton', 'Elia-Martell', 'Alliser-Thorne', 'Rafford', 'Arya-Stark', 'Boros-Blount', 'Styr', 'Hot-Pie', 'Loras-Tyrell', 'Tansy', 'Ilyn-Payne', 'Mandon-Moore', 'Stannis-Baratheon', 'Alester-Florent', 'Sharna', 'Jaime-Lannister', 'Balon-Swann', 'Harma', 'Tywin-Lannister', 'Pycelle', 'Harwin', 'Vargo-Hoat', 'Robert-Baratheon', 'Donal-Noye', 'Edric-Storm', 'Janos-Slynt', 'Tickler'}
{'Roose-Bolton', 'Jaime-Lannister', 'Brienne-of-Tarth', 'Walton', 'Qyburn'}
{'Missandei', 'Jhiqui', 'Daenerys-Targaryen', 'Irri', 'Jorah-Mormont'}
{'Arstan', 'Jhiqui', 'Belwas', 'Daenerys-Targaryen', 'Jorah-Mormont'}
{'Drogo', 'Daario-Naharis', 'Daenerys-Targaryen', 'Irri', 'Jorah-Mormont'}

3-Components:

{'Alla-Tyrell', 'Megga-Tyrell', 'Elinor-Tyrell', 'Sansa-Stark'}
{'Amory-Lorch', 'Beric-Dondarrion', 'Marq-Piper', 'Rickon-Stark', 'Willas-Tyrell', 'Lark', 'Mace-Tyrell', 'Mully', 'Greenbeard', 'Robert-Arryn', 'Arthur-Dayne', 'Khorane-Sathmantes', 'Gendry', 'Walder-Frey', 'Jeor-Mormont', 'Olenna-Redwyne', 'Tanda-Stokeworth', 'Barristan-Selmy', 'Dunsen', 'Dontos-Hollard', 'Rhaegar-Targaryen', 'Meera-Reed', 'Tommen-Baratheon', 'Eddard-Stark', 'Grigg', 'Meryn-Trant', 'Mance-Rayder', 'Gerold-Hightower', 'Brienne-of-Tarth', 'Ryk', 'Osmund-Kettleblack', 'Small-Paul', 'Kevan-Lannister', 'Ryman-Frey', 'Cersei-Lannister', 'Margaery-Tyrell', 'Jojen-Reed', 'Robb-Stark', 'Rickard-Karstark', 'Alayaya', 'Qyburn', 'Davos-Seaworth', 'Bowen-Marsh', 'Clydas',

'Eddison-Tollett', 'Doran-Martell', 'Viserys-Targaryen', 'Joffrey-Baratheon',
 'Bronn', 'Ygritte', 'Varys', 'Val', 'Anguy', 'Jarl', 'Edmure-Tully', 'Shireen-
 Baratheon', 'Aemon-Targaryen-(Maester-Aemon)', 'Brynden-Tully', 'Pypar', 'Tom-
 of-Sevenstreams', 'Jon-Snow', 'Hobb', 'Owen', 'Podrick-Payne', 'Symon-Silver-
 Tongue', 'Roose-Bolton', 'Elia-Martell', 'Daenerys-Targaryen', 'Alliser-Thorne',
 'Desmond-Grell', 'Rafford', 'Stevron-Frey', 'Arya-Stark', 'Boros-Blount',
 'Loras-Tyrell', 'Ilyn-Payne', 'Marillion', 'Balon-Swann', 'Tywin-Lannister',
 'Pycelle', 'Donal-Noye', 'Dacey-Mormont', 'Edric-Storm', 'Tickler', 'Porridge',
 'Cleos-Frey', 'Theon-Greyjoy', 'Pylos', 'Gilly', 'Varamyr', 'Craster', 'Bran-
 Stark', 'Petyr-Baelish', 'Renly-Baratheon', 'Shae', 'Sansa-Stark', 'Selyse-
 Florent', 'Benjen-Stark', 'Myrcella-Baratheon', 'Grenn', 'Spare-Boot', 'Dywen',
 'Elmar-Frey', 'Salladhor-Saan', 'Hareth-(Moles-Town)', 'Axell-Florent', 'Errok',
 'Aerys-II-Targaryen', 'Walton', 'Kegs', 'Polliver', 'Gregor-Clegane', 'Tormund',
 'Lollys-Stokeworth', 'Oberyn-Martell', 'Hodor', 'Maege-Mormont', 'Lothar-Frey',
 'Hoster-Tully', 'Garlan-Tyrell', 'Jon-Arryn', 'Lamprey', 'Addam-Marbrand',
 'Lothor-Brune', 'Catelyn-Stark', 'Samwell-Tarly', 'Rattleshirt', 'Lem',
 'Melisandre', 'Tyrion-Lannister', 'Chett', 'Devan-Seaworth', 'Qhorin-Halfhand',
 'Denys-Mallister', 'Sandor-Clegane', 'Bedwyck', 'Satin', 'Brandon-Stark',
 'Balon-Greyjoy', 'Lysa-Arryn', 'Paxter-Redwyne', 'Ellaria-Sand', 'Roslin-Frey',
 'Jon-Umber-(Greatjon)', 'Dalla', 'High-Septon-(Tyrions)', 'Cotter-Pyke',
 'Thoros-of-Myr', 'Othell-Yarwyck', 'Aegon-Frey-(son-of-Stevron)', 'Jorah-
 Mormont', 'Baelor-I-Targaryen', 'Styr', 'Hot-Pie', 'Mycah', 'Tansy', 'Alester-
 Florent', 'Stannis-Baratheon', 'Mandon-Moore', 'Sharna', 'Jaime-Lannister',
 'Harma', 'Harwin', 'Robert-Baratheon', 'Vargo-Hoat', 'Janos-Slynt'}
 {'Missandei', 'Arstan', 'Daario-Naharis', 'Daenerys-Targaryen', 'Mero',
 'Ben-Plumm', 'Irri', 'Jorah-Mormont', 'Drogo', 'Jhiqui', 'Belwas'}

2-Components:

{'Jalabhar-Xho', 'Gyles-Rosby', 'Addam-Marbrand'}
 {'Alla-Tyrell', 'Megga-Tyrell', 'Elinor-Tyrell', 'Sansa-Stark'}
 {'Sallor', 'Prendahl-na-Ghezn', 'Daario-Naharis'}
 {'Gorne', 'Jon-Snow', 'Gendel'}
 {'Robett-Glover', 'Helman-Tallhart', 'Randyll-Tarly'}
 {'Cleon', 'Daenerys-Targaryen', 'Ghael'}
 {'Daenerys-Targaryen', 'Rakharo', 'Aggo', 'Jhogo'}
 {'Jonothor-Darry', 'Lewyn-Martell', 'Oswell-Whent'}
 {'Amory-Lorch', 'Beric-Dondarrion', 'Marq-Piper', 'Rickon-Stark',
 'Willas-Tyrell', 'Lark', 'Mace-Tyrell', 'Olyvar-Frey', 'Mully', 'Greenbeard',
 'Robert-Arryn', 'Rorge', 'Belwas', 'Arthur-Dayne', 'Drogo', 'Khorane-
 Sathmantes', 'Gendry', 'Gawen-Westerling', 'Walder-Frey', 'Mathis-Rowan', 'Jeor-
 Mormont', 'Irri', 'Olenna-Redwyne', 'Tanda-Stokeworth', 'Barristan-Selmy',
 'Dunsen', 'Guncer-Sunglass', 'Rhaegar-Targaryen', 'Meera-Reed', 'Dontos-
 Hollard', 'Alerie-Hightower', 'Tommen-Baratheon', 'Eddard-Stark', 'Grigg',
 'Meryn-Trant', 'Mance-Rayder', 'Orell', 'Gerold-Hightower', 'Brienne-of-Tarth',
 'Ryk', 'Brella', 'Butterbumps', 'Osmund-Kettleblack', 'Small-Paul', 'Kevan-
 Lannister', 'Ryman-Frey', 'Cersei-Lannister', 'Nan', 'Margaery-Tyrell', 'Jojen-
 Reed', 'Robb-Stark', 'Rickard-Karstark', 'Missandei', 'Merrett-Frey', 'Alayaya',
 'Qyburn', 'Davos-Seaworth', 'Bowen-Marsh', 'Clydas', 'Eddison-Tollett', 'Doran-'

Martell', 'Viserys-Targaryen', 'Kraznys-mo-Nakloz', 'Joffrey-Baratheon',
 'Bronn', 'Ygritte', 'Varys', 'Jhiqui', 'Lancel-Lannister', 'Jack-Bulwer',
 'Oswell-Kettleblack', 'Val', 'Anguy', 'Jarl', 'Edmure-Tully', 'Shireen-
 Baratheon', 'Jon-Umber-(Smalljon)', 'Aemon-Targaryen-(Maester-Aemon)', 'Brynden-
 Tully', 'Pypar', 'Tom-of-Sevenstreams', 'Vyman', 'Jon-Snow', 'Walder-Rivers',
 'Hobb', 'Owen', 'Podrick-Payne', 'Symon-Silver-Tongue', 'Thoren-Smallwood',
 'Roose-Bolton', 'Elia-Martell', 'Coldhands', 'Daenerys-Targaryen', 'Alliser-
 Thorne', 'Desmond-Grell', 'Rafford', 'Stevron-Frey', 'Arya-Stark', 'Boros-
 Blount', 'Loras-Tyrell', 'Bannen', 'Ilyn-Payne', 'Marillion', 'Oznak-zo-Pahl',
 'Mero', 'Balon-Swann', 'Utherydes-Wayn', 'Tywin-Lannister', 'Pycelle', 'Donal-
 Noye', 'Dacey-Mormont', 'Edric-Storm', 'Wendel-Manderly', 'Tickler', 'Porridge',
 'Cleos-Frey', 'Theon-Greyjoy', 'Pylos', 'Gilly', 'Varamyr', 'Craster', 'Bran-
 Stark', 'Zollo', 'Torrhen-Stark', 'Petyr-Baelish', 'Shagwell', 'Renly-
 Baratheon', 'Shae', 'Sansa-Stark', 'Selyse-Florent', 'Benjen-Stark', 'Myrcella-
 Baratheon', 'Grenn', 'Spare-Boot', 'Ben-Plumm', 'Aegon-I-Targaryen', 'Dywen',
 'Weeper', 'Grey-Worm', 'Elmar-Frey', 'Salladhor-Saan', 'Dick-Follard',
 'Husband', 'Axell-Florent', 'Errrok', 'Aerys-II-Targaryen', 'Walton', 'Kegs',
 'Polliver', 'Hareth-(Moles-Town)', 'Rhaella-Targaryen', 'Jeyne-Westerling',
 'Daario-Naharis', 'Lewyn-Martell', 'Gregor-Clegane', 'Tormund', 'Illyrio-
 Mopatis', 'Lollys-Stokeworth', 'Andrew-Estermont', 'Oberyn-Martell', 'Mag-Mar-
 Tun-Doh-Weg', 'Hodor', 'Maege-Mormont', 'Willem-Lannister', 'Lothar-Frey',
 'Arstan', 'Hoster-Tully', 'Garlan-Tyrell', 'Jon-Arryn', 'Lamprey', 'Addam-
 Marbrand', 'Lothor-Brune', 'Catelyn-Stark', 'Sybell-Spicer', 'Robin-Flint',
 'Samwell-Tarly', 'Rattleshirt', 'Lem', 'Clubfoot-Karl', 'Melisandre', 'Tyrion-
 Lannister', 'Chett', 'Devan-Seaworth', 'Qhorin-Halfhand', 'Denys-Mallister',
 'Sandor-Clegane', 'Bedwyck', 'Satin', 'Big-Boil', 'Brandon-Stark', 'Dirk',
 'Balon-Greyjoy', 'Lysa-Arryn', 'Paxter-Redwyne', 'Ellaria-Sand', 'Roslin-Frey',
 'Jon-Umber-(Greatjon)', 'Tysha', 'Dalla', 'Robin-Ryger', 'High-
 Septon-(Tyrions)', 'Cotter-Pyke', 'Thoros-of-Myr', 'Othell-Yarwyck', 'Aegon-
 Frey-(son-of-Stevron)', 'Jorah-Mormont', 'Baelor-I-Targaryen', 'Styr', 'Hot-
 Pie', 'Ramsay-Snow', 'Mycah', 'Martyn-Lannister', 'Tansy', 'Alester-Florent',
 'Stannis-Baratheon', 'Mandon-Moore', 'Raynald-Westerling', 'Sharna', 'Jaime-
 Lannister', 'Harma', 'Harwin', 'Rollam-Westerling', 'Robert-Baratheon', 'Petyr-
 Frey', 'Vargo-Hoat', 'Janos-Slynt'}

1-Components:

{'Amory-Lorch', 'Beric-Dondarrion', 'Willas-Tyrell', 'Greenbeard',
 'Robert-Arryn', 'Rorge', 'Drogo', 'Moon-Boy', 'Watt', 'Khorane-Sathmantes',
 'Gendry', 'Walder-Frey', 'Mathis-Rowan', 'Jonthor-Darry', 'Joramun', 'Olenna-
 Redwyne', 'Ottyn-Wythers', 'Tanda-Stokeworth', 'Merrit', 'Dunsen', 'Dontos-
 Hollard', 'Rhaegar-Targaryen', 'Meera-Reed', 'Guncer-Sunglass', 'Alerie-
 Hightower', 'Grigg', 'Meryn-Trant', 'Mance-Rayder', 'Gerold-Hightower', 'Ryk',
 'Brella', 'Butterbumps', 'Osmund-Kettleblack', 'Ryman-Frey', 'Rakharo', 'Jojen-
 Reed', 'Rickard-Karstark', 'Merrett-Frey', 'Grazdan-mo-Eraz', 'Alayaya',
 'Qyburn', 'Davos-Seaworth', 'Doran-Martell', 'Eddison-Tollett', 'Joffrey-
 Baratheon', 'Bronn', 'Ygritte', 'Blane', 'Jhiqui', 'Jack-Bulwer', 'Oswell-
 Kettleblack', 'Edmure-Tully', 'Shireen-Baratheon', 'Jon-Umber-(Smalljon)',
 'Aemon-Targaryen-(Maester-Aemon)', 'Pypar', 'Cleon', 'Jon-Snow', 'Walder-'

Rivers', 'Hobb', 'Podrick-Payne', 'Symon-Silver-Tongue', 'Owen', 'Thoren-Smallwood', 'Byam-Flint', 'Smiling-Knight', 'Jason-Mallister', 'Daenerys-Targaryen', 'Zei', 'Desmond-Grell', 'Stevron-Frey', 'Boros-Blount', 'Wynton-Stout', 'Loras-Tyrell', 'Marillion', 'Mero', 'Tywin-Lannister', 'Urswyck', 'Wendel-Manderly', 'Rossart', 'Tickler', 'Azor-Ahai', 'Porridge', 'Cleos-Frey', 'Ravella-Swann', 'Gilly', 'Varamyr', 'Craster', 'Shae', 'Sansa-Stark', 'Spare-Boot', 'Ben-Plumm', 'Dywen', 'Maegor-I-Targaryen', 'Weeper', 'Elinor-Tyrell', 'Elmar-Frey', 'Hareth-(Moles-Town)', 'Aemon-Targaryen-(Dragonknight)', 'Husband', 'Axell-Florent', 'Aerys-II-Targaryen', 'Walton', 'Rodrik-Cassel', 'Polliver', 'Kegs', 'Daario-Naharis', 'Gregor-Clegane', 'Lollys-Stokeworth', 'Tion-Frey', 'Oberry-Martell', 'Mag-Mar-Tun-Doh-Weg', 'Maege-Mormont', 'Willem-Lannister', 'Lothar-Frey', 'Ghael', 'Sallor', 'Hoster-Tully', 'Garlan-Tyrell', 'Jon-Arryn', 'Daeron-I-Targaryen', 'Catelyn-Stark', 'Sybell-Spicer', 'Robin-Flint', 'Rattleshirt', 'Lem', 'Tyrion-Lannister', 'Kella', 'Oswell-Whent', 'Jalabhar-Xho', 'Qhorin-Halfhand', 'Sandor-Clegane', 'Bedwyck', 'Satin', 'Brandon-Stark', 'Lysa-Arryn', 'Ellaria-Sand', 'Tysha', 'Dalla', 'Aegon-Targaryen-(son-of-Rhaegar)', 'Robin-Ryger', 'Megga-Tyrell', 'Thoros-of-Myr', 'Gorne', 'Gendel', 'Aegon-Frey-(son-of-Stevron)', 'Baelor-I-Targaryen', 'Jorah-Mormont', 'Mycah', 'Martyn-Lannister', 'Tansy', 'Perwyn-Frey', 'Stannis-Baratheon', 'Sharna', 'Jaime-Lannister', 'Harma', 'Robert-Baratheon', 'Vargo-Hoat', 'Rolph-Spicer', 'Lark', 'Marq-Piper', 'Olyvar-Frey', 'Rickon-Stark', 'Mace-Tyrell', 'Mully', 'Robett-Glover', 'Prendahl-na-Ghezn', 'Belwas', 'Arthur-Dayne', 'Luwin', 'Gawen-Westerling', 'Tytos-Blackwood', 'Jeor-Mormont', 'Irri', 'Leonette-Fossoway', 'Barristan-Selmy', 'Gyles-Rosby', 'Patchface', 'Jack-B-Lucky', 'Arson', 'Tommen-Baratheon', 'Eddard-Stark', 'Galbart-Glover', 'Orell', 'Brienne-of-Tarth', 'Cressen', 'Small-Paul', 'Kevan-Lannister', 'Cersei-Lannister', 'Nan', 'Margaery-Tyrell', 'Missandei', 'Robb-Stark', 'Bowen-Marsh', 'Clydas', 'Viserys-Targaryen', 'Kraznys-mo-Nakloz', 'Anders-Yronwood', 'Galyeon-of-Cuy', 'Ballabar', 'Tristifer-IV-Mudd', 'Yohn-Royce', 'Varys', 'Halder', 'Lancel-Lannister', 'Val', 'Anguy', 'Jarl', 'Softfoot', 'Munda', 'Vyman', 'Brynden-Tully', 'Tom-of-Sevenstreams', 'Randyll-Tarly', 'Roose-Bolton', 'Helman-Tallhart', 'Nage', 'Elia-Martell', 'Coldhands', 'Alliser-Thorne', 'Rafford', 'Arya-Stark', 'Jhogo', 'Gerald-Gower', 'Bannen', 'Todder', 'Ragwyle', 'Ilyn-Payne', 'Oznak-zo-Pahl', 'Utherydes-Wayn', 'Balon-Swann', 'Pycelle', 'Donal-Noye', 'Dacey-Mormont', 'Edric-Storm', 'Theon-Greyjoy', 'Pylos', 'Bran-Stark', 'Grisel', 'Zollo', 'Torren-Stark', 'Shagwell', 'Petyr-Baelish', 'Renly-Baratheon', 'Biter', 'Selyse-Florent', 'Benjen-Stark', 'Myrcella-Baratheon', 'Greenn', 'Pate-(Old)', 'Chataya', 'Aegon-I-Targaryen', 'Grey-Worm', 'Salladhor-Saan', 'Dick-Follard', 'Errok', 'Rhaella-Targaryen', 'Jeyne-Westerling', 'Lewyn-Martell', 'Aggo', 'Tormund', 'Illyrio-Mopatis', 'Andrew-Estermont', 'Hodor', 'Arstan', 'Lamprey', 'Addam-Marbrand', 'Lothor-Brunne', 'Samwell-Tarly', 'Groleo', 'Melisandre', 'Clubfoot-Karl', 'Chett', 'Devan-Seaworth', 'Denys-Mallister', 'Mordane', 'Big-Boil', 'Dirk', 'Balon-Greyjoy', 'Paxter-Redwyne', 'Roslin-Frey', 'Jon-Umber-(Greatjon)', 'Rhaegel-Targaryen', 'Alla-Tyrell', 'High-Septon-(Tyrions)', 'Cotter-Pyke', 'Othell-Yarwyck', 'Styr', 'Hot-Pie', 'Brenett', 'Ramsay-Snow', 'Mallador-Locke', 'Mandon-Moore', 'Raynald-Westerling', 'Alester-Florent', 'Mad-Huntsman', 'Frenken', 'Harwin', 'Simon-Toyne', 'Rollam-Westerling', 'Joanna-Lannister', 'Petyr-Frey', 'Janos-Slynt'}

All found K-Components in Book 4

7-Components:

```
{'Aurane-Waters', 'Mace-Tyrell', 'Tommen-Baratheon', 'Harys-Swyft',
'Taena-of-Myr', 'Loras-Tyrell', 'Osmund-Kettleblack', 'Stannis-Baratheon',
'Cersei-Lannister', 'Jaime-Lannister', 'Margaery-Tyrell', 'Pycelle', 'Tywin-
Lannister', 'Qyburn', 'Gyles-Rosby', 'Tyrion-Lannister', 'Orton-Merryweather'}
```

6-Components:

```
{'Aurane-Waters', 'Mace-Tyrell', 'Tommen-Baratheon', 'Harys-Swyft',
'Eddard-Stark', 'Joffrey-Baratheon', 'Gregor-Clegane', 'Osney-Kettleblack',
'Sandor-Clegane', 'Brienne-of-Tarth', 'Taena-of-Myr', 'Loras-Tyrell', 'Petyr-
Baelish', 'Osmund-Kettleblack', 'Renly-Baratheon', 'Kevan-Lannister', 'Stannis-
Baratheon', 'Sansa-Stark', 'Cersei-Lannister', 'Jaime-Lannister', 'Margaery-
Tyrell', 'Tywin-Lannister', 'Pycelle', 'Qyburn', 'Catelyn-Stark', 'Robert-
Baratheon', 'Gyles-Rosby', 'Tyrion-Lannister', 'Orton-Merryweather'}
```

5-Components:

```
{'Mace-Tyrell', 'Joffrey-Baratheon', 'Sandor-Clegane', 'Petyr-Baelish',
'Renly-Baratheon', 'Sansa-Stark', 'High-Sparrow', 'Jocelyn-Swyft', 'Gyles-
Rosby', 'Orton-Merryweather', 'Aurane-Waters', 'Tommen-Baratheon', 'Harys-
Swyft', 'Eddard-Stark', 'Gregor-Clegane', 'Brienne-of-Tarth', 'Taena-of-Myr',
'Loras-Tyrell', 'Osmund-Kettleblack', 'Kevan-Lannister', 'Stannis-Baratheon',
'Cersei-Lannister', 'Jaime-Lannister', 'Margaery-Tyrell', 'Tywin-Lannister',
'Pycelle', 'Qyburn', 'Catelyn-Stark', 'Robert-Baratheon', 'Osney-Kettleblack',
'Tyrion-Lannister', 'Dorcus'}
```

```
{'Cersei-Lannister', 'Ilyn-Payne', 'Gregor-Clegane', 'Meryn-Trant',
'Rafford', 'Dunsen'}
```

4-Components:

```
{'Aeron-Greyjoy', 'Euron-Greyjoy', 'Asha-Greyjoy', 'Baelor-Blacktyde',
'Balon-Greyjoy', 'Victarion-Greyjoy'}
{'Mace-Tyrell', 'Petyr-Baelish', 'Renly-Baratheon', 'Sansa-Stark',
'Walder-Frey', 'Myrcella-Baratheon', 'Olenna-Redwyne', 'Jocelyn-Swyft', 'Gyles-
Rosby', 'Aerys-II-Targaryen', 'Orton-Merryweather', 'Dunsen', 'Dontos-Hollard',
'Rhaegar-Targaryen', 'Aurane-Waters', 'Tommen-Baratheon', 'Harys-Swyft',
'Eddard-Stark', 'Gregor-Clegane', 'Meryn-Trant', 'Brienne-of-Tarth', 'Taena-of-
Myr', 'Osmund-Kettleblack', 'Kevan-Lannister', 'Ryman-Frey', 'Cersei-Lannister',
'Margaery-Tyrell', 'Jon-Arryn', 'Addam-Marbrand', 'Qyburn', 'Catelyn-Stark',
'Tyrion-Lannister', 'Emmon-Frey', 'Joffrey-Baratheon', 'Varys', 'Sandor-
Clegane', 'Lancel-Lannister', 'Lysa-Arryn', 'Edmure-Tully', 'High-Sparrow',
'Brynden-Tully', 'Lyle-Crakehall', 'Rafford', 'Boros-Blount', 'Loras-Tyrell',
'Ilyn-Payne', 'Stannis-Baratheon', 'Jaime-Lannister', 'Tywin-Lannister',
'Pycelle', 'Robert-Baratheon', 'Osney-Kettleblack', 'Dorcus'}
{Anya-Waynwood', 'Symond-Templeton', 'Gilwood-Hunter', 'Horton-
Redfort', 'Benedar-Belmore', 'Petyr-Baelish', 'Yohn-Royce'}
```

```

['Robert-Arryn', 'Sansa-Stark', 'Nestor-Royce', 'Petyr-Baelish', 'Lysa-
Arryn', 'Yohn-Royce']
{'Mollander', 'Alleras', 'Roone', 'Armen', 'Pate-(novice)', 'Leo-
Tyrell'}
{'Myrcella-Baratheon', 'Sylva-Santagar', 'Andrey-Dalt', 'Gerold-Dayne',
'Tyene-Sand', 'Obara-Sand', 'Nymeria-Sand', 'Garin-(orphan)', 'Arianne-Martell',
'Arys-Oakheart', 'Doran-Martell', 'Areo-Hotah', 'Oberyn-Martell'}

```

3-Components:

```

{'Beric-Dondarrion', 'Mace-Tyrell', 'Amerei-Frey', 'Robert-Arryn',
'Moon-Boy', 'Walder-Frey', 'Tallad', 'Olenna-Redwyne', 'Jocelyn-Swyft', 'Nestor-
Royce', 'Gyles-Rosby', 'Orton-Merryweather', 'Dunsen', 'Dontos-Hollard',
'Rhaegar-Targaryen', 'Aurane-Waters', 'Tommen-Baratheon', 'Harys-Swyft',
'Eddard-Stark', 'Colemon', 'Meryn-Trant', 'Brienne-of-Tarth', 'Taena-of-Myr',
'Osmund-Kettleblack', 'Dareon', 'Kevan-Lannister', 'Ryman-Frey', 'Cersei-
Lannister', 'Margaery-Tyrell', 'Robb-Stark', 'Qyburn', 'Doran-Martell', 'Emmon-
Frey', 'Genna-Lannister', 'Joffrey-Baratheon', 'Bronn', 'Yohn-Royce', 'Varys',
'Lancel-Lannister', 'Gretchel', 'Edmure-Tully', 'Myranda-Royce', 'Aemon-
Targaryen-(Maester-Aemon)', 'Brynden-Tully', 'Jon-Snow', 'Walder-Rivers',
'Podrick-Payne', 'Randyll-Tarly', 'Lyle-Crakehall', 'Rafford', 'Arya-Stark',
'Boros-Blount', 'Loras-Tyrell', 'Ilyn-Payne', 'Balon-Swann', 'Pycelle', 'Tywin-
Lannister', 'Osney-Kettleblack', 'Dorcus', 'Gilly', 'Bran-Stark', 'Petyr-
Baelish', 'Renly-Baratheon', 'Hyle-Hunt', 'Sansa-Stark', 'Myrcella-Baratheon',
'Maegor-I-Targaryen', 'Elinor-Tyrell', 'Aerys-II-Targaryen', 'Anya-Waynwood',
'Symond-Templeton', 'Gregor-Clegane', 'Falyse-Stokeworth', 'Meribald', 'Mya-
Stone', 'Harrold-Hardyng', 'Balman-Byrch', 'Gilwood-Hunter', 'Jon-Arryn',
'Edwyn-Frey', 'Osfryd-Kettleblack', 'Addam-Marbrand', 'Lothor-Brone', 'Catelyn-
Stark', 'Samwell-Tarly', 'Tyrion-Lannister', 'Jalabhar-Xho', 'Horton-Redfort',
'Lambert-Turnberry', 'Sandor-Clegane', 'Dick-Crabb', 'Lysa-Arryn', 'Daven-
Lannister', 'High-Sparrow', 'Blue-Bard', 'Alla-Tyrell', 'Megga-Tyrell', 'Thoros-
of-Myr', 'Benedar-Belmore', 'Stannis-Baratheon', 'Jaime-Lannister', 'Pia',
'Robert-Baratheon', 'Maddy'}
{'Pyg', 'Shagwell', 'Brienne-of-Tarth', 'Timeon'}
{'Marwyn', 'Mollander', 'Alleras', 'Roone', 'Pate-(novice)', 'Armen',
'Samwell-Tarly', 'Leo-Tyrell'}
{'Caleotte', 'Sylva-Santagar', 'Obara-Sand', 'Gerold-Dayne', 'Andrey-
Dalt', 'Tyene-Sand', 'Myrcella-Baratheon', 'Garin-(orphan)', 'Arianne-Martell',
'Arys-Oakheart', 'Doran-Martell', 'Oberyn-Martell', 'Areo-Hotah', 'Nymeria-
Sand'}
{'Elder-Brother', 'Narbert', 'Brienne-of-Tarth', 'Meribald'}
{'Aeron-Greyjoy', 'Euron-Greyjoy', 'Asha-Greyjoy', 'Baelor-Blacktyde',
'Balon-Greyjoy', 'Victarion-Greyjoy'}

```

2-Components:

```

{'Talea', 'Brusco', 'Brea', 'Arya-Stark'}
{'Ternesio-Terys', 'Denyo-Terys', 'Arya-Stark'}
{'Waif', 'Kindly-Man', 'Arya-Stark'}
{'Rodrik-Harlaw', 'Quellon-Greyjoy', 'Asha-Greyjoy', 'Red-Oarsman',

```

'Gorold-Goodbrother', 'Theon-Greyjoy', 'Victarion-Greyjoy', 'Ralf-Stonehouse',
 'Sigfryd-Harlaw', 'Murenmure', 'Balon-Greyjoy', 'Meldred-Merlyn', 'Baelor-
 Blacktyde', 'Dunstan-Drumm', 'Aeron-Greyjoy', 'Euron-Greyjoy', 'Hotho-Harlaw',
 'Nute', 'Humfrey-Hewett'}
 {'Mellei', 'Arianne-Martell', 'Morra'}
 {'Terrance-Lynderly', 'Gyles-Grafton', 'Robert-Arryn'}
 {'Brienne-of-Tarth', 'Willow-Heddle', 'Gendry'}
 {'Quhuru-Mo', 'Xhondo', 'Samwell-Tarly'}
 {'Beric-Dondarrion', 'Rickon-Stark', 'Mace-Tyrell', 'Josmyn-Peckledon',
 'Amerei-Frey', 'Robert-Arryn', 'Moon-Boy', 'Melara-Hetherspoon', 'Walder-Frey',
 'Tallad', 'Jeor-Mormont', 'Lyonel-Corbray', 'Jocelyn-Swyft', 'Olenna-Redwyne',
 'Nestor-Royce', 'Illifer', 'Gyles-Rosby', 'Orton-Merryweather', 'Dunsen',
 'Dontos-Hollard', 'Rhaegar-Targaryen', 'Aurane-Waters', 'Harys-Swyft', 'Tommen-
 Baratheon', 'Eddard-Stark', 'Colemon', 'Meryn-Trant', 'Brienne-of-Tarth',
 'Taena-of-Myr', 'Trystane-Martell', 'Osmund-Kettleblack', 'Dareon', 'Kevan-
 Lannister', 'Ryman-Frey', 'Cersei-Lannister', 'Margaery-Tyrell',
 'Pate-(novice)', 'Robb-Stark', 'Qyburn', 'Shadrich', 'Clydas', 'Doran-Martell',
 'Eddison-Tollett', 'Caleotte', 'Emmon-Frey', 'Genna-Lannister', 'Kojja-Mo',
 'Tyene-Sand', 'Joffrey-Baratheon', 'Bronn', 'Yohn-Royce', 'Varys', 'Lancel-
 Lannister', 'Gretchel', 'Edmure-Tully', 'Myranda-Royce', 'Narbert', 'Aemon-
 Targaryen-(Maester-Aemon)', 'Brynden-Tully', 'Pypar', 'Jon-Snow', 'Walder-
 Rivers', 'Armen', 'Podrick-Payne', 'Randyll-Tarly', 'Lyle-Crakehall',
 'Garin-(orphan)', 'Arya-Stark', 'Rafford', 'Boros-Blount', 'Pyg', 'Arianne-
 Martell', 'Arys-Oakheart', 'Loras-Tyrell', 'Ilyn-Payne', 'Marwyn', 'Marillion',
 'Balon-Swann', 'Tywin-Lannister', 'Quentyn-Martell', 'Pycelle', 'Osney-
 Kettleblack', 'Dorcus', 'Cedra', 'Gilly', 'Craster', 'Bran-Stark', 'Zollo',
 'Leo-Tyrell', 'Petyr-Baelish', 'Renly-Baratheon', 'Shagwell', 'Hyle-Hunt',
 'Mollander', 'Sansa-Stark', 'Myrcella-Baratheon', 'Grenn', 'Maegor-I-Targaryen',
 'Forley-Prester', 'Elinor-Tyrell', 'Senelle', 'Aerys-II-Targaryen', 'Anya-
 Waynwood', 'Sylva-Santagar', 'Symond-Templeton', 'Andrey-Dalt', 'Gerold-Dayne',
 'Elder-Brother', 'Falyse-Stokeworth', 'Gregor-Clegane', 'Creighton-Longbough',
 'Meribald', 'Mya-Stone', 'Roone', 'Harrold-Hardyng', 'Areo-Hotah', 'Oberry-
 Martell', 'Gilwood-Hunter', 'Balman-Byrch', 'Jon-Arryn', 'Edwyn-Frey', 'Osfryd-
 Kettleblack', 'Addam-Marbrand', 'Lothor-Brone', 'Catelyn-Stark', 'Samwell-
 Tarly', 'Daemon-Sand', 'Melisandre', 'Tyrion-Lannister', 'Jalabhar-Xho',
 'Horton-Redfort', 'Lambert-Turnberry', 'Sandor-Clegane', 'Mariya-Darry', 'Dick-
 Crabb', 'Paxter-Redwyne', 'Lysa-Arryn', 'Roslin-Frey', 'Daven-Lannister',
 'Obara-Sand', 'High-Sparrow', 'Lyn-Corbray', 'Blue-Bard', 'Alla-Tyrell', 'Megga-
 Tyrell', 'High-Septon-(Tyrions)', 'Thoros-of-Myr', 'Alleras', 'Timeon',
 'Benedar-Belmore', 'Maggy', 'Nymeria-Sand', 'Stannis-Baratheon', 'Jaime-
 Lannister', 'Pia', 'Robert-Baratheon', 'Maddy', 'Vargo-Hoat'}

1-Components:

{'Beric-Dondarrion', 'Willas-Tyrell', 'Willow-Heddle', 'Robert-Arryn',
 'Raynard', 'Franklyn-Fowler', 'Moon-Boy', 'Gendry', 'Walder-Frey', 'Mathis-
 Rowan', 'Cedric-Payne', 'Lyonel-Corbray', 'Olenna-Redwyne', 'Quhuru-Mo', 'Garth-
 Tyrell', 'Theobald', 'Tanda-Stokeworth', 'Orton-Merryweather', 'Humfrey-Hewett',
 'Dunsen', 'Dontos-Hollard', 'Rhaegar-Targaryen', 'Aurane-Waters', 'Harys-Swyft',

'Talbert-Serry', 'Mervyn-Trant', 'Ralf-Stonehouse', 'Taena-of-Myr', 'Trystane-Martell', 'Brella', 'Osmund-Kettleblack', 'Ryman-Frey', 'Pate-(novice)', 'Qyburn', 'Doran-Martell', 'Eddison-Tollett', 'Nute', 'Genna-Lannister', 'Kojja-Mo', 'Tyene-Sand', 'Joffrey-Baratheon', 'Bronn', 'Gerold-Grafton', 'Morra', 'Edmure-Tully', 'Myranda-Royce', 'Narbert', 'Aemon-Targaryen-(Maester-Aemon)', 'Pypar', 'Jon-Snow', 'Walder-Rivers', 'Jaehaerys-I-Targaryen', 'Gormond-Goodbrother', 'Podrick-Payne', 'Jeyne-Farman', 'Lyle-Crakehall', 'Garin-(orphan)', 'Boros-Blount', 'Arys-Oakheart', 'Karyl-Vance', 'Waif', 'Loras-Tyrell', 'Rennifer-Longwaters', 'Marillion', 'Marwyn', 'Tywin-Lannister', 'Quentyn-Martell', 'Syrio-Forel', 'Osney-Kettleblack', 'Tickler', 'Cleos-Frey', 'Asha-Greyjoy', 'Joy-Hill', 'Gilly', 'Meralyn', 'Craster', 'Elbert-Arryn', 'Meldred-Merlyn', 'Shae', 'Hyle-Hunt', 'Mollander', 'Sansa-Stark', 'Emmond', 'Maegor-I-Targaryen', 'Walgrave', 'Elinor-Tyrell', 'Senelle', 'Aerys-II-Targaryen', 'Anya-Waywood', 'Sylva-Santagar', 'Andrey-Dalt', 'Gerold-Dayne', 'Terrance-Lynderly', 'Gregor-Clegane', 'Gorold-Goodbrother', 'Lollys-Stokeworth', 'Denys-Darklyn', 'Mya-Stone', 'Areo-Hotah', 'Obryen-Martell', 'Baelor-Blacktyde', 'Brea', 'Balman-Byrch', 'Garlan-Tyrell', 'Jon-Arryn', 'Gilwood-Hunter', 'Osfryd-Kettleblack', 'Elys-Waywood', 'Catelyn-Stark', 'Sybell-Spicer', 'Hugo-Vance', 'Tyrion-Lannister', 'Ternesio-Terys', 'Jalabhar-Xho', 'Horton-Redfort', 'Qhorin-Halfhand', 'Sandor-Clegane', 'Puckens', 'Kindly-Man', 'Mariya-Darry', 'Dick-Crabb', 'Lysa-Arryn', 'Obara-Sand', 'Clement-Piper', 'Lyn-Corbray', 'Dalla', 'Rus', 'Aegon-Targaryen-(son-of-Rhaegar)', 'Blue-Bard', 'Megga-Tyrell', 'Wylis-Manderly', 'Thoros-of-Myr', 'Alleras', 'Timeon', 'Rosey', 'Baelor-I-Targaryen', 'Maggy', 'Stannis-Baratheon', 'Jaime-Lannister', 'Pia', 'Sawane-Botley', 'Robert-Baratheon', 'Aeron-Greyjoy', 'Vargo-Hoat', 'Ronnet-Connington', 'Rickon-Stark', 'Mark-Mullendore', 'Mace-Tyrell', 'Josmyn-Peckledon', 'Amerei-Frey', 'Rugen', 'Sigfryd-Harlaw', 'Sarella-Sand', 'Greydon-Goodbrother', 'Luwin', 'Melara-Hetherspoon', 'Tallad', 'Jeor-Mormont', 'Jocelyn-Swift', 'Nestor-Royce', 'Illifer', 'Barristan-Selmy', 'Gyles-Rosby', 'Talea', 'Tommen-Baratheon', 'Eddard-Stark', 'Shitmouth', 'Colemon', 'Brusco', 'Aegon-V-Targaryen', 'Brienne-of-Tarth', 'Dareon', 'Kevan-Lannister', 'Cersei-Lannister', 'Nan', 'Margaery-Tyrell', 'Robb-Stark', 'Mellei', 'Shadrich', 'Clydas', 'Caleotte', 'Emmon-Frey', 'Anders-Yronwood', 'Yohn-Royce', 'Quill', 'Varys', 'Lancel-Lannister', 'Gretchel', 'Manfrey-Martell', 'Torbert', 'Lanna-(Happy-Port)', 'Brynden-Tully', 'Euron-Greyjoy', 'Armen', 'Hotho-Harlaw', 'Lewys-Piper', 'Randyll-Tarly', 'Roose-Bolton', 'Rodrik-Harlaw', 'Elia-Martell', 'Rafford', 'Arya-Stark', 'Eustace-Brune', 'Pyg', 'Arianne-Martell', 'Lucas-Codd', 'Ilyn-Payne', 'Balon-Swann', 'Pycelle', 'Shortear', 'Dorcus', 'Cedra', 'Tristifer-Botley', 'Theon-Greyjoy', 'Bran-Stark', 'Zollo', 'Petyr-Baelish', 'Shagwell', 'Renly-Baratheon', 'Leo-Tyrell', 'Torwold-Brownooth', 'Biter', 'Pate-(whipping-boy)', 'Ralf-(Limper)', 'Myrcella-Baratheon', 'Grenn', 'Xhondo', 'Dunstan-Drumm', 'Bonifer-Hasty', 'Forley-Prester', 'Jeyne-Heddle', 'Symond-Templeton', 'Jeyne-Westerling', 'Elder-Brother', 'Falyse-Stokeworth', 'Victarion-Greyjoy', 'Creighton-Longbough', 'Meribald', 'Timoth', 'Murenmore', 'Roone', 'Harrold-Hardyng', 'Edwyn-Frey', 'Addam-Marbrand', 'Lothor-Brune', 'Samwell-Tarly', 'Daemon-Sand', 'Bellegere-Others', 'Melisandre', 'Gyles-Grafton', 'Denyo-Terys', 'Quellon-Greyjoy', 'Lambert-Turnberry', 'Balon-Greyjoy', 'Paxter-Redwyne', 'Roslin-Frey', 'Garrett-Paege', 'Daven-Lannister',

'High-Sparrow', 'Clarence-Crabb', 'Unella', 'Alla-Tyrell', 'High-Septon-(Tyrions)', 'Yorko-Terys', 'Umma', 'Gormon-Tyrell', 'Red-Oarsman', 'Benedar-Belmore', 'Ricasso', 'Owen-Merryweather', 'Nymeria-Sand', 'Harras-Harlaw', 'Alys-Arryn', 'Maddy'}

All found K-Components in Book 5

6-Components:

{'Skahaz-mo-Kandaq', 'Missandei', 'Hizdahr-zo-Loraq', 'Daario-Naharis', 'Daenerys-Targaryen', 'Irri', 'Grey-Worm', 'Jhiqui', 'Barristan-Selmy'}
{'Haldon', 'Yandry', 'Rolly-Duckfield', 'Aegon-Targaryen-(son-of-Rhaegar)', 'Jon-Connington', 'Ysilla', 'Tyrion-Lannister', 'Lemore'}

5-Components:

{'Skahaz-mo-Kandaq', 'Hizdahr-zo-Loraq', 'Daario-Naharis', 'Daenerys-Targaryen', 'Aggo', 'Jhogo', 'Jhiqui', 'Belwas', 'Rakharo', 'Missandei', 'Irri', 'Quentyn-Martell', 'Grey-Worm', 'Barristan-Selmy'}
{'Skahaz-mo-Kandaq', 'Daenerys-Targaryen', 'Hizdahr-zo-Loraq', 'Reznak-mo-Reznak', 'Galazza-Galare', 'Barristan-Selmy'}
{'Roose-Bolton', 'Stannis-Baratheon', 'Ramsay-Snow', 'Eddard-Stark', 'Wyman-Manderly', 'Theon-Greyjoy'}
{'Cersei-Lannister', 'Ilyn-Payne', 'Gregor-Clegane', 'Meryn-Trant', 'Rafford', 'Dunsen'}
{'Haldon', 'Yandry', 'Rolly-Duckfield', 'Aegon-Targaryen-(son-of-Rhaegar)', 'Jon-Connington', 'Ysilla', 'Tyrion-Lannister', 'Lemore'}

4-Components:

{'Skahaz-mo-Kandaq', 'Archibald-Yronwood', 'Hizdahr-zo-Loraq', 'Daario-Naharis', 'Galazza-Galare', 'Daenerys-Targaryen', 'Aggo', 'Jhogo', 'Hero', 'Belwas', 'Jhiqui', 'Reznak-mo-Reznak', 'Spotted-Cat', 'Tattered-Prince', 'Gerris-Drinkwater', 'Meris', 'Rakharo', 'Missandei', 'Cletus-Yronwood', 'Goghor-the-Giant', 'Khazz', 'Irri', 'Ben-Plumm', 'Quentyn-Martell', 'Bloodbeard', 'Grey-Worm', 'Caggo', 'Barristan-Selmy', 'Belaquo', 'Grolio'}
{'Roose-Bolton', 'Stannis-Baratheon', 'Eddard-Stark', 'Wyman-Manderly', 'Theon-Greyjoy', 'Barbrey-Dustin', 'Ramsay-Snow', 'Jeyne-Poole'}
{'Mully', 'Tormund', 'Mance-Rayder', 'Gilly', 'Satin', 'Val', 'Stannis-Baratheon', 'Selyse-Florent', 'Shireen-Baratheon', 'Eddison-Tollett', 'Dalla', 'Aemon-Targaryen-(Maester-Aemon)', 'Jon-Snow', 'Bowen-Marsh', 'Clydas', 'Samwell-Tarly', 'Hobb', 'Axell-Florent', 'Melisandre'}
{'Haldon', 'Rhaegar-Targaryen', 'Tommen-Baratheon', 'Daenerys-Targaryen', 'Varys', 'Illyrio-Mopatis', 'Ysilla', 'Kevan-Lannister', 'Stannis-Baratheon', 'Cersei-Lannister', 'Myrcella-Baratheon', 'Yandry', 'Rolly-Duckfield', 'Tywin-Lannister', 'Aegon-Targaryen-(son-of-Rhaegar)', 'Robert-Baratheon', 'Jon-Connington', 'Barristan-Selmy', 'Tyrion-Lannister', 'Lemore'}
{'Theon-Greyjoy', 'Ramsay-Snow', 'Damon-Dance-for-Me', 'Skinner', 'Sour-Alyn'}
{'Cersei-Lannister', 'Ilyn-Payne', 'Gregor-Clegane', 'Meryn-Trant',

'Rafford', 'Dunsen'}

3-Components:

{'Meera-Reed', 'Jojen-Reed', 'Bran-Stark', 'Coldhands', 'Hodor'}
{'Jack-Bulwer', 'Hairy-Hal', 'Jon-Snow', 'Garth-Greyfeather'}
{'Tristifer-Botley', 'Qarl-the-Maid', 'Hagen', 'Asha-Greyjoy'}
{'Skahaz-mo-Kandaq', 'Archibald-Yronwood', 'Hizdahr-zo-Loraq',
'Grazhar', 'Hero', 'Belwas', 'Jhiqui', 'Qezza', 'Tattered-Prince', 'Meris',
'Goghor-the-Giant', 'Khazz', 'Irri', 'Ben-Plumm', 'SteelSkin', 'Grey-Worm',
'Caggo', 'Barristan-Selmy', 'Belaquo', 'Daario-Naharis', 'Galazza-Galare',
'Daenerys-Targaryen', 'Aggo', 'Jhogo', 'Reznak-mo-Reznak', 'Spotted-Cat',
'Gerris-Drinkwater', 'Rakharo', 'Missandei', 'Cletus-Yronwood', 'Quentyn-
Martell', 'Bloodbeard', 'Groleo'}
{'Rickon-Stark', 'Asha-Greyjoy', 'Mully', 'Theon-Greyjoy', 'Gilly',
'Clayton-Suggs', 'Bran-Stark', 'Cellador', 'Godry-Farring', 'Sansa-Stark',
'Selyse-Florent', 'Barbrey-Dustin', 'Holly', 'Weeper', 'Hareth-(Moles-Town)',
'Axell-Florent', 'Eddard-Stark', 'Tormund', 'Mance-Rayder', 'Alys-Karstark',
'Alysane-Mormont', 'Robb-Stark', 'Bowen-Marsh', 'Clydas', 'Samwell-Tarly',
'Eddison-Tollett', 'Melisandre', 'Hop-Robin', 'Devan-Seaworth', 'Narbert-
Grandison', 'Satin', 'Leathers', 'Emmett', 'Val', 'Toregg', 'Shireen-Baratheon',
'Dalla', 'Aemon-Targaryen-(Maester-Aemon)', 'Squirrel', 'Wun-Weg-Wun-Dar-Wun',
'Jon-Snow', 'Hobb', 'Othell-Yarwyck', 'Roose-Bolton', 'Patrek-of-Kings-
Mountain', 'Justin-Massey', 'Mors-Umber', 'Malegorn', 'Arya-Stark', 'Sigorn',
'Ramsay-Snow', 'Hother-Umber', 'Jeyne-Poole', 'Tycho-Nestoris', 'Stannis-
Baratheon', 'Wyman-Manderly', 'Arnolf-Karstark', 'Rowan', 'Robert-Baratheon',
'Brus-Buckler', 'Janos-Slynt'}
{'Haldon', 'Rhaegar-Targaryen', 'Tommen-Baratheon', 'Daenerys-
Targaryen', 'Joffrey-Baratheon', 'Qavo-Nogarys', 'Varys', 'Illyrio-Mopatis',
'Arianne-Martell', 'Tyrion-Lannister', 'Ysilla', 'Areo-Hotah', 'Ellaria-Sand',
'Kevan-Lannister', 'Stannis-Baratheon', 'Cersei-Lannister', 'Myrcella-
Baratheon', 'Obara-Sand', 'Moqorro', 'Benerro', 'Balon-Swann', 'Yandry', 'Rolly-
Duckfield', 'Tywin-Lannister', 'Aegon-Targaryen-(son-of-Rhaegar)', 'Robert-
Baratheon', 'Doran-Martell', 'Jon-Connington', 'Barristan-Selmy', 'Aerys-II-
Targaryen', 'Lemore'}
{'Grunt', 'Theon-Greyjoy', 'Ramsay-Snow', 'Yellow-Dick', 'Damon-Dance-
for-Me', 'Skinner', 'Sour-Alyn'}
{'Nurse', 'Penny', 'Daenerys-Targaryen', 'Ben-Plumm', 'Jorah-Mormont',
'Sweets', 'Yezzan-zo-Qaggaz', 'Tyrion-Lannister'}
{'Hareth-(Moles-Town)', 'Arron', 'Satin', 'Emrick'}
{'Moelle', 'Cersei-Lannister', 'Unella', 'Scolera'}
{'Tyrion-Lannister', 'Ben-Plumm', 'Kasporio', 'Tybero-Istarion'}
{'Cersei-Lannister', 'Ilyn-Payne', 'Gregor-Clegane', 'Meryn-Trant',
'Rafford', 'Dunsen'}

2-Components:

{'Meera-Reed', 'Jojen-Reed', 'Bran-Stark', 'Coldhands', 'Leaf', 'Hodor',
'Brynden-Rivers'}
{'Waif', 'Kindly-Man', 'Arya-Stark'}

{'Jeor-Mormont', 'Jon-Snow', 'Qhorin-Halfhand', 'Donal-Noye'}
{'Jack-Bulwer', 'Hairy-Hal', 'Jon-Snow', 'Garth-Greyfeather'}
{'Pypar', 'Todder', 'Grenn'}
{'Asha-Greyjoy', 'Sybelle-Glover', 'Grimtongue', 'Qarl-the-Maid',
'Cromm', 'Tristifer-Botley', 'Hagen', 'Lorren'}
{'Skahaz-mo-Kandaq', 'Rickon-Stark', 'Hizdahr-zo-Loraq', 'Mace-Tyrell',
'Grazhar', 'Mully', 'Robett-Glover', 'Qavo-Nogarys', 'Miklaz', 'Drogo',
'Belwas', 'Yellow-Dick', 'Cellador', 'Godry-Farring', 'Meris', 'Rory',
'Franklyn-Flowers', 'Khazz', 'Goghhor-the-Giant', 'Yandry', 'Irri', 'Walda-
Frey-(daughter-of-Merrett)', 'Sour-Alyn', 'Barbrey-Dustin', 'Steelsskin',
'Holly', 'Joramun', 'Barristan-Selmy', 'Patchface', 'Leona-Woolfield', 'Dunsen',
'Rhaegar-Targaryen', 'Tommen-Baratheon', 'Eddard-Stark', 'Hosteen-Frey', 'Harry-
Strickland', 'Meryn-Trant', 'Mance-Rayder', 'Brienne-of-Tarth', 'Kevan-
Lannister', 'Rakharo', 'Missande', 'Cersei-Lannister', 'Walder-Frey-(son-of-
Merrett)', 'Robb-Stark', 'Scar', 'Bloodbeard', 'Qyburn', 'Davos-Seaworth',
'Bowen-Marsh', 'Clydas', 'Doran-Martell', 'Eddison-Tollett', 'Caleotte', 'Hop-
Robin', 'Haldon', 'Mezzara', 'Viserys-Targaryen', 'Penny', 'Emrick', 'Tyene-
Sand', 'Joffrey-Baratheon', 'Ygritte', 'Hero', 'Varys', 'Jhiqui', 'Ysilla',
'Leathers', 'Val', 'Tattered-Prince', 'Toregg', 'Shireen-Baratheon', 'Aemon-
Targaryen-(Maester-Aemon)', 'Squirrel', 'Cleon', 'Jon-Snow', 'Euron-Greyjoy',
'Hobb', 'Jon-Connington', 'Owen', 'Robert-Strong', 'Lemore', 'Roose-Bolton',
'Mors-Umber', 'Patrek-of-Kings-Mountain', 'Randyll-Tarly', 'Elia-Martell',
'Daenerys-Targaryen', 'Malegorn', 'Jhogo', 'Arya-Stark', 'Rafford', 'Boros-
Blount', 'Sigorn', 'Arianne-Martell', 'Arys-Oakheart', 'Hother-Umber', 'Arron',
'Yezzan-zo-Qaggaz', 'Spotted-Cat', 'Ilyn-Payne', 'Balon-Swann', 'Tywin-
Lannister', 'Quentyn-Martell', 'Pycelle', 'Garth-(Wolfs-Den)', 'Brus-Buckler',
'Emmett', 'Asha-Greyjoy', 'Theon-Greyjoy', 'Gilly', 'Clayton-Suggs', 'Bran-
Stark', 'Qezza', 'Xaro-Xhoan-Daxos', 'Roger-Ryswell', 'Damon-Dance-for-Me',
'Sansa-Stark', 'Selyse-Florent', 'Myrcella-Baratheon', 'Ben-Plumm', 'Rolly-
Duckfield', 'Weeper', 'Grey-Worm', 'Salladhor-Saan', 'Hareth-(Moles-Town)',
'Axell-Florent', 'Aerys-II-Targaryen', 'Rodrik-Cassel', 'Kedry', 'Grunt',
'Gerold-Dayne', 'Daario-Naharis', 'Moelle', 'Gregor-Clegane', 'Aggo', 'Tormund',
'Victarion-Greyjoy', 'Richard-Horpe', 'Illyrio-Mopatis', 'Reznak-mo-Reznak',
'Areo-Hotah', 'Alys-Karstark', 'Ben-Bones', 'Oberyn-Martell', 'Gerris-
Drinkwater', 'Alysane-Mormont', 'Moqorro', 'Cletus-Yronwood', 'Myles-Toyne',
'Scolera', 'Sybell-Spicer', 'Samwell-Tarly', 'Rattleshirt', 'Grolio',
'Melisandre', 'Tyrion-Lannister', 'Walder-Frey-(son-of-Jammos)', 'Archibald-
Yronwood', 'Devan-Seaworth', 'Frenya', 'Fulk', 'Sweets', 'Narbert-Grandison',
'Denys-Mallister', 'Marlon-Manderly', 'Satin', 'Balon-Greyjoy', 'Ellaria-Sand',
'Aenys-Frey', 'Obara-Sand', 'Benerro', 'Benethon-Scales', 'High-Sparrow',
'Dalla', 'Skinner', 'Theomore', 'Unella', 'Aegon-Targaryen-(son-of-Rhaegar)',
'Wun-Weg-Wun-Dar-Wun', 'Caggo', 'Kasporio', 'Belaquo', 'Cotter-Pyke', 'Othell-
Yarwyck', 'Nurse', 'Justin-Massey', 'Galazza-Galare', 'Jorah-Mormont', 'Ramsay-
Snow', 'Willam-Dustin', 'Wick-Whittlestick', 'Jeyne-Poole', 'Tycho-Nestoris',
'Nymeria-Sand', 'Stannis-Baratheon', 'Rodrik-Sparr', 'Jaime-Lannister', 'Wyman-
Manderly', 'Arnolf-Karstark', 'Aeron-Greyjoy', 'Robert-Baratheon', 'Tybero-
Istarion', 'Rowan', 'Janos-Slynt'}
{'Daenerys-Targaryen', 'Marselen', 'Symon-Stripeback'}

1-Components:

{'Skahaz-mo-Kandaq', 'Hizdahr-zo-Loraq', 'Grazhar', 'Miklaz', 'Drogo', 'Godry-Farring', 'Franklyn-Flowers', 'Goghor-the-Giant', 'Yandry', 'Harl', 'Haggan', 'Walda-Frey-(daughter-of-Merrett)', 'Joramun', 'SteelSkin', 'Malaquo-Maegyr', 'Holly', 'Leona-Woolfield', 'Lyanna-Stark', 'Dunsen', 'Rhaegar-Targaryen', 'Meera-Reed', 'Bartimus', 'Brandon-Norrey', 'Harys-Swyft', 'Meryn-Trant', 'Mance-Rayder', 'Harwood-Stout', 'Trystane-Martell', 'Rakharo', 'Walder-Frey-(son-of-Merrett)', 'Jojen-Reed', 'Rickard-Karstark', 'Scar', 'Bloodbeard', 'Qyburn', 'Rhaegar-Frey', 'Davos-Seaworth', 'Daeron-II-Targaryen', 'Doran-Martell', 'Eddison-Tollett', 'Hop-Robin', 'Marghaz-zo-Loraq', 'Garth-Greyfeather', 'Torghen-Flint', 'Haldon', 'Tyene-Sand', 'Joffrey-Baratheon', 'Ygritte', 'Jhiqui', 'Jack-Bulwer', 'Leathers', 'Tom-Barleycorn', 'Toregg', 'Wulfe', 'Shireen-Baratheon', 'Aemon-Targaryen-(Maester-Aemon)', 'Squirrel', 'Cleon', 'Pypar', 'Jon-Snow', 'Hobb', 'Owen', 'Robert-Strong', 'Lemore', 'Daenerys-Targaryen', 'Malegorn', 'Boros-Blount', 'Sigorn', 'Arys-Oakheart', 'Waif', 'Yezzan-zo-Qaggaz', 'Spotted-Cat', 'Dick-Straw', 'Quentyn-Martell', 'Tywin-Lannister', 'Garth-(Wolfs-Den)', 'Old-Bill-Bone', 'Osney-Kettleblack', 'Black-Balaq', 'Azor-Ahai', 'Asha-Greyjoy', 'Grimtongue', 'Gilly', 'Varamyr', 'Raymun-Redbeard', 'Xaro-Xhoan-Daxos', 'Roger-Ryswell', 'Damon-Dance-for-Me', 'Shae', 'Bump', 'Sansa-Stark', 'Ben-Plumm', 'Rolly-Duckfield', 'Hairy-Hal', 'Weeper', 'Hareth-(Moles-Town)', 'Axell-Florent', 'Pono', 'Aerys-II-Targaryen', 'Rodrik-Cassel', 'Kedry', 'Grunt', 'Daario-Naharis', 'Gerold-Dayne', 'Gregor-Clegane', 'Wex-Pyke', 'Areo-Hotah', 'Oberyn-Martell', 'Ben-Bones', 'Godric-Borrell', 'Ghael', 'Maege-Mormont', 'Brea', 'Alysane-Mormont', 'Gerris-Drinkwater', 'Cletus-Yronwood', 'Hoster-Blackwood', 'Torgon-Greyiron', 'Scolera', 'Brynden-Rivers', 'Catelyn-Stark', 'Sybell-Spicer', 'Ashara-Dayne', 'Rattleshirt', 'Tyrion-Lannister', 'Frenya', 'Archibald-Yronwood', 'Qhorin-Halfhand', 'Narbert-Grandison', 'Luton', 'Satin', 'Kindly-Man', 'Lorren', 'Kerwin', 'Ellaria-Sand', 'Denzo-Dhan', 'Tysha', 'Obara-Sand', 'Skinner', 'Theomore', 'Dalla', 'Nyessos-Vhassar', 'Aegon-Targaryen-(son-of-Rhaegar)', 'Wun-Weg-Wun-Dar-Wun', 'Caggo', 'Wylis-Manderly', 'Draqaz', 'Nurse', 'Jorah-Mormont', 'Ulmer', 'William-Foxglove', 'Willam-Dustin', 'Wick-Whittlestick', 'Dryn', 'Tycho-Nestoris', 'Stannis-Baratheon', 'Rodrik-Sparr', 'Jaime-Lannister', 'Arnolf-Karstark', 'Robert-Baratheon', 'Tybero-Istarion', 'Aeron-Greyjoy', 'Symon-Stripeback', 'Ronnet-Connington', 'Rickon-Stark', 'Mace-Tyrell', 'Corliss-Penny', 'Robett-Glover', 'Josmyn-Peckledon', 'Mully', 'Qavo-Nogarys', 'Belwas', 'Yellow-Dick', 'Cellador', 'Meris', 'Tytos-Blackwood', 'Rory', 'Khazz', 'Jeor-Mormont', 'Irri', 'Quaithe', 'Sour-Alyn', 'Barbrey-Dustin', 'Barristan-Selmy', 'Patchface', 'Hugh-Hungerford', 'Tommen-Baratheon', 'Eddard-Stark', 'Hosteen-Frey', 'Harry-Strickland', 'Brusco', 'Galbart-Glover', 'Brienne-of-Tarth', 'Widower', 'Kevan-Lannister', 'Morgo', 'Cersei-Lannister', 'Missandei', 'Margaery-Tyrell', 'Robb-Stark', 'Larraq', 'Willow-Witch-eye', 'Shrouded-Lord', 'Bowen-Marsh', 'Clydas', 'Caleotte', 'Mezzara', 'Viserys-Targaryen', 'Penny', 'Emrick', 'Hero', 'Varys', 'Lancel-Lannister', 'Hagen', 'Ysilla', 'Val', 'Tattered-Prince', 'Brynden-Tully', 'Euron-Greyjoy', 'Jon-Connington', 'Randyll-Tarly', 'Patrek-of-Kings-Mountain', 'Roose-Bolton', 'Mors-Umber', 'Elia-Martell', 'Coldhands', 'Alliser-Thorne', 'Qarl-the-Maid', 'Jhogo', }

```
'Arya-Stark', 'Rafford', 'Arianne-Martell', 'Hother-Umber', 'Marselen',
'Todder', 'Arron', 'Ilyn-Payne', 'Ezzara', 'Jonos-Bracken', 'Rhaenys-
Targaryen-(daughter-of-Rhaegar)', 'Balon-Swann', 'Sybelle-Glover', 'Pycelle',
'Lyanna-Mormont', 'Rolfe-the-Dwarf', 'Yurkhaz-zo-Yunzak', 'Gerrick-Kingsblood',
'Donal-Noye', 'Brus-Buckler', 'Emmett', 'Marya-Seaworth', 'Tristifer-Botley',
'Theon-Greyjoy', 'Clayton-Suggs', 'Bran-Stark', 'Qezza', 'Petyr-Baelish',
'Ralf-(Lipper)', 'Selyse-Florent', 'Benjen-Stark', 'Myrcella-Baratheon',
'Grenn', 'Aegon-I-Targaryen', 'Grey-Worm', 'Salladhor-Saan', 'Moelle', 'Aggo',
'Tormund', 'Victarion-Greyjoy', 'Richard-Horpe', 'Illyrio-Mopatis', 'Reznak-mo-
Reznak', 'Alys-Karstark', 'Leaf', 'Hodor', 'Barsena', 'Moqorro', 'Arthor-
Karstark', 'Myles-Toyne', 'Thistle', 'Roose-Ryswell', 'Samwell-Tarly', 'Grolio',
'Melisandre', 'Walder-Frey-(son-of-Jammos)', 'Devan-Seaworth', 'Fulk', 'Cromm',
'Sweets', 'Lysono-Maar', 'Longwater-Pyke', 'Denys-Mallister', 'Marlon-Manderly',
'Soren-Shieldbreaker', 'Balon-Greyjoy', 'Jon-Umber-(Greatjon)', 'Aenys-Frey',
'Benethon-Scales', 'Benerro', 'High-Sparrow', 'Unella', 'Kasporio', 'Belaquo',
'Cotter-Pyke', 'Othell-Yarwyck', 'Justin-Massey', 'Galazza-Galare', 'Lucifer-
Long', 'Eroeh', 'Camarron', 'Ramsay-Snow', 'Jeyne-Poole', 'Nymeria-Sand',
'Wyman-Manderly', 'Rowan', 'Jared-Frey', 'Janos-Slynt'}
```

Local Clustering Nel nostro contenuto i valori di local clustering sono una sorta di valore di centralità. Infatti, rappresentano quanti “buchi” ci sono intorno ad un nodo. Se un nodo ha molti buchi, vuol dire che è centrale nella storia e il suo valore di local clustering sarà molto basso.

Nel nostro caso ci aspettiamo che le comparse avranno local clustering elevato, mentre i personaggi principali avranno local clustering coefficients molto bassi.

```
[ ]: def find_most_important_characters(d, n):
    """
    Trova i n elementi con valore più basso in un dizionario, considerando solo
    i valori maggiori di 0 e minori di 1.

    :param d: Dizionario da cui estrarre gli elementi.
    :param n: Numero di elementi da restituire.
    :return: Lista dei n elementi con i valori più bassi.
    """

    # Filtrare il dizionario per includere solo i valori > 0 e < 1
    filtered_items = {k: v for k, v in d.items() if 0 < v < 1}

    # Ordinare il dizionario filtrato per valore in ordine crescente e prendere
    # i primi n elementi
    least_n = sorted(filtered_items.items(), key=lambda item: item[1])[:n]

    return least_n
```

```
[ ]: def find_least_important_characters(d, n):
    """
    Trova i n elementi con valore più basso in un dizionario, considerando solo
    i valori maggiori di 0 e minori di 1.
```

```

:param d: Dizionario da cui estrarre gli elementi.
:param n: Numero di elementi da restituire.
:return: Lista dei n elementi con i valori più bassi.
"""

# Filtrare il dizionario per includere solo i valori > 0 e < 1
filtered_items = {k: v for k, v in d.items() if 0 < v < 1}

# Ordinare il dizionario filtrato per valore in ordine crescente e prendere
# i primi n elementi
least_n = sorted(filtered_items.items(), key=lambda item: item[1],
reverse=True)[:n]

return least_n

```

```

[ ]: def plot_clustering_histogram(char_clustering, title = ""):
    """
    Stampa un istogramma del coefficiente di clustering locale per i personaggi.

    :param char_clustering: Lista di tuple (personaggio, valore di clustering).
    """

    # Separare i personaggi e i loro valori
    characters, values = zip(*char_clustering)

    # Creare l'istogramma
    plt.figure(figsize=(10, 5))
    bars = plt.bar(characters, values, color='blue', alpha=0.7)

    # Aggiungere i valori sopra le barre
    for bar in bars:
        yval = bar.get_height()
        plt.text(bar.get_x() + bar.get_width()/2, yval, round(yval, 2),
                 ha='center', va='bottom', fontsize=10)

    # Personalizzare il grafico
    plt.title(title, color="red")
    plt.xlabel('Nodes')
    plt.ylabel('Local Clustering Coefficient')
    plt.xticks(rotation=45, ha='right')
    plt.grid(axis='y', alpha=0.75)
    plt.tight_layout() # Migliora la disposizione

    # Mostrare l'istogramma
    plt.show()

```

```
[ ]: clustering_1 = nx.clustering(G_1, nodes=None, weight=None)
clustering_2 = nx.clustering(G_2, nodes=None, weight=None)
clustering_3 = nx.clustering(G_3, nodes=None, weight=None)
clustering_4 = nx.clustering(G_4, nodes=None, weight=None)
clustering_5 = nx.clustering(G_5, nodes=None, weight=None)
clustering_all = nx.clustering(G_all, nodes=None, weight=None)

[ ]: important_char_1 = find_most_important_characters(clustering_1, 10)
not_important_char_1 = find_least_important_characters(clustering_1, 10)

important_char_2 = find_most_important_characters(clustering_2, 10)
not_important_char_2 = find_least_important_characters(clustering_2, 10)

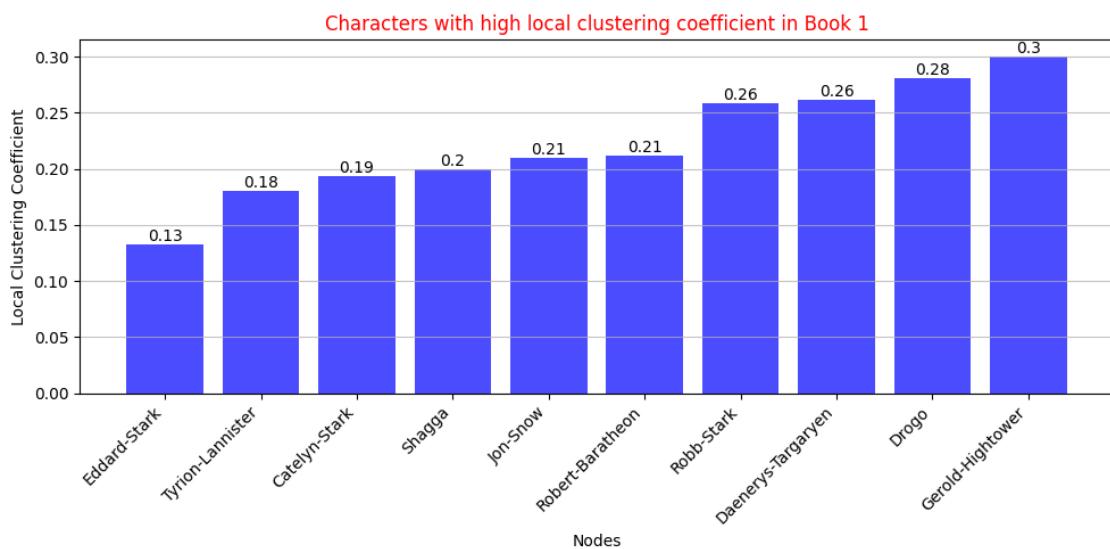
important_char_3 = find_most_important_characters(clustering_3, 10)
not_important_char_3 = find_least_important_characters(clustering_3, 10)

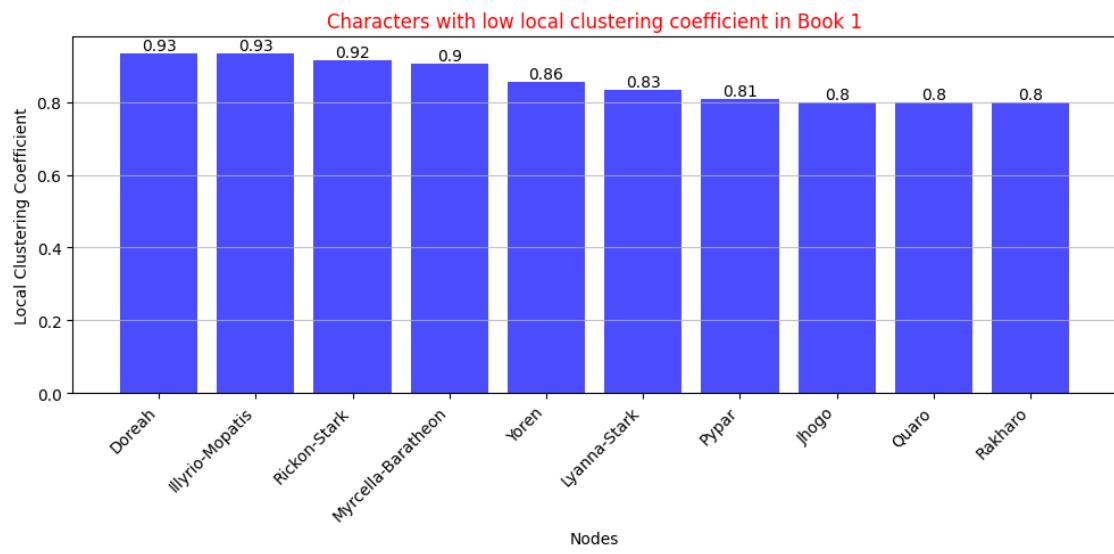
important_char_4 = find_most_important_characters(clustering_4, 10)
not_important_char_4 = find_least_important_characters(clustering_4, 10)

important_char_5 = find_most_important_characters(clustering_5, 10)
not_important_char_5 = find_least_important_characters(clustering_5, 10)

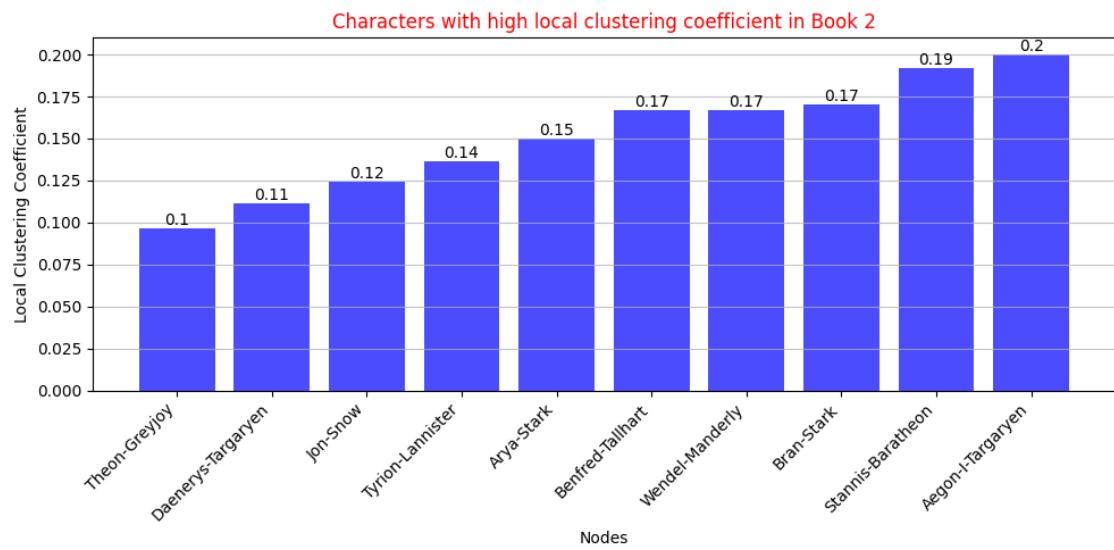
important_char_all = find_most_important_characters(clustering_all, 10)
not_important_char_all = find_least_important_characters(clustering_all, 10)

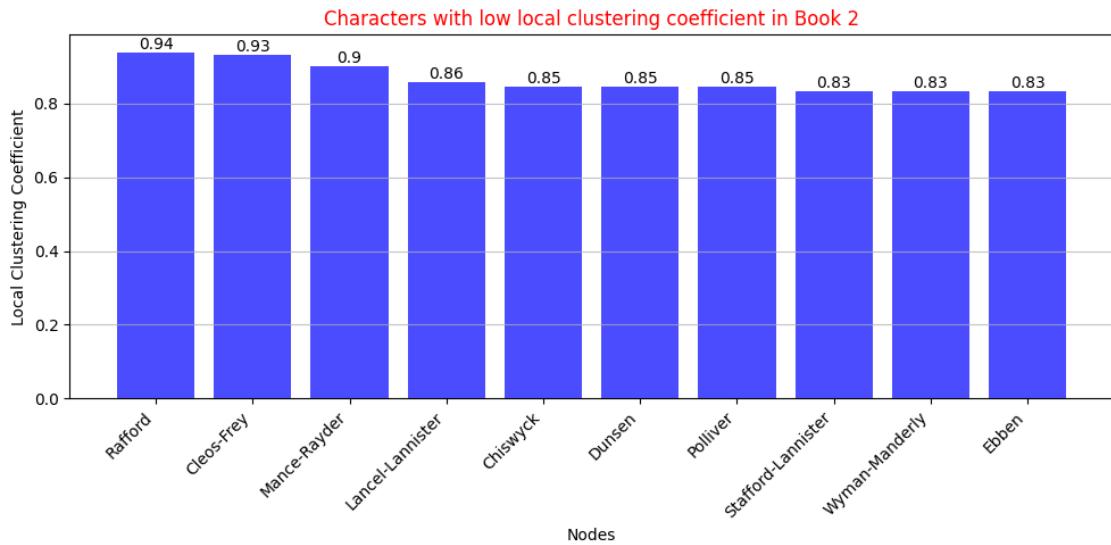
[ ]: plot_clustering_histogram(important_char_1, title="Characters with high local clustering coefficient in Book 1")
plot_clustering_histogram(not_important_char_1, title="Characters with low local clustering coefficient in Book 1")
```



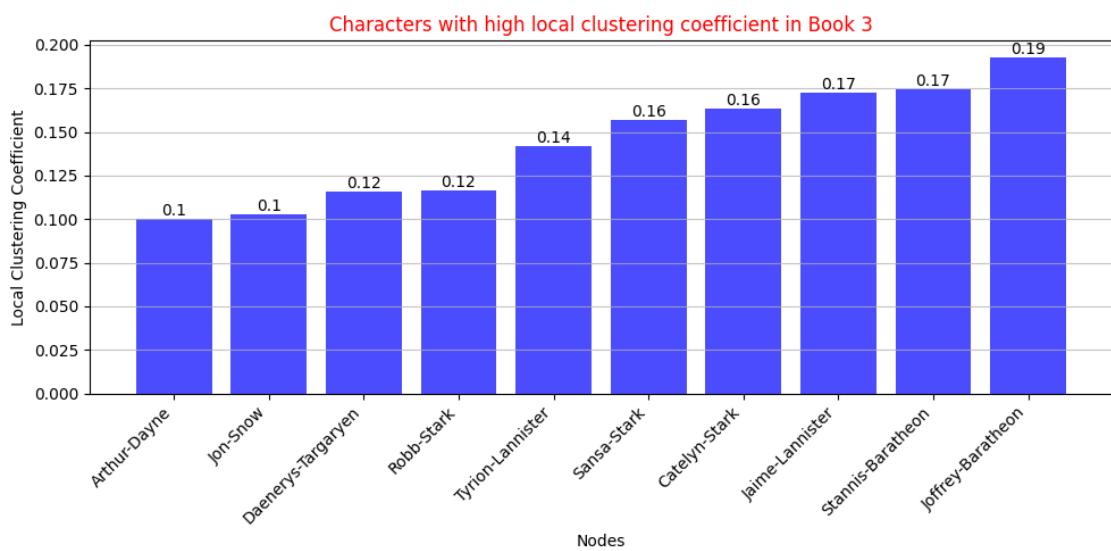


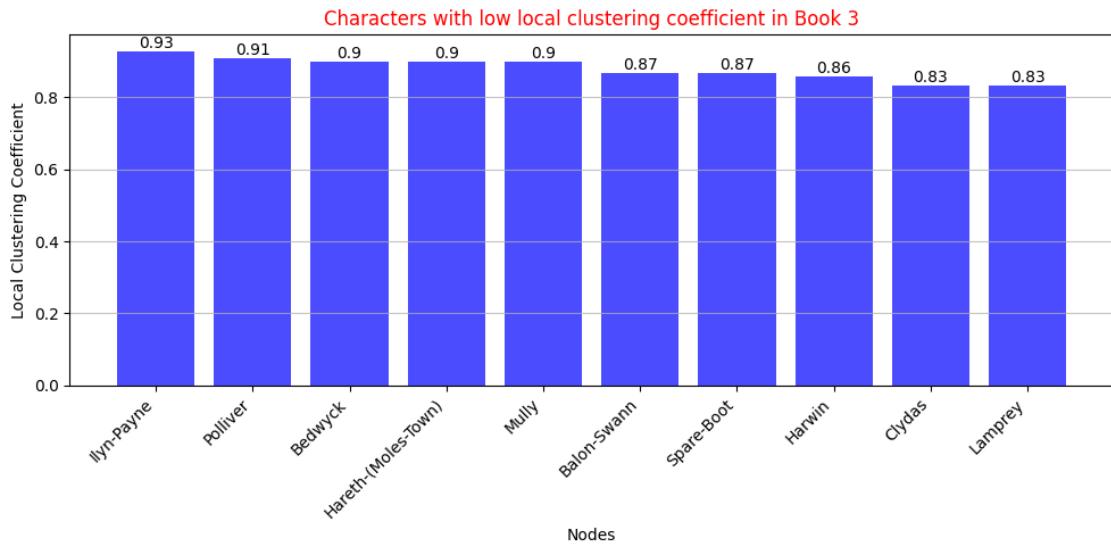
```
[ ]: plot_clustering_histogram(important_char_2, title="Characters with high local clustering coefficient in Book 2")
plot_clustering_histogram(not_important_char_2, title="Characters with low local clustering coefficient in Book 2")
```



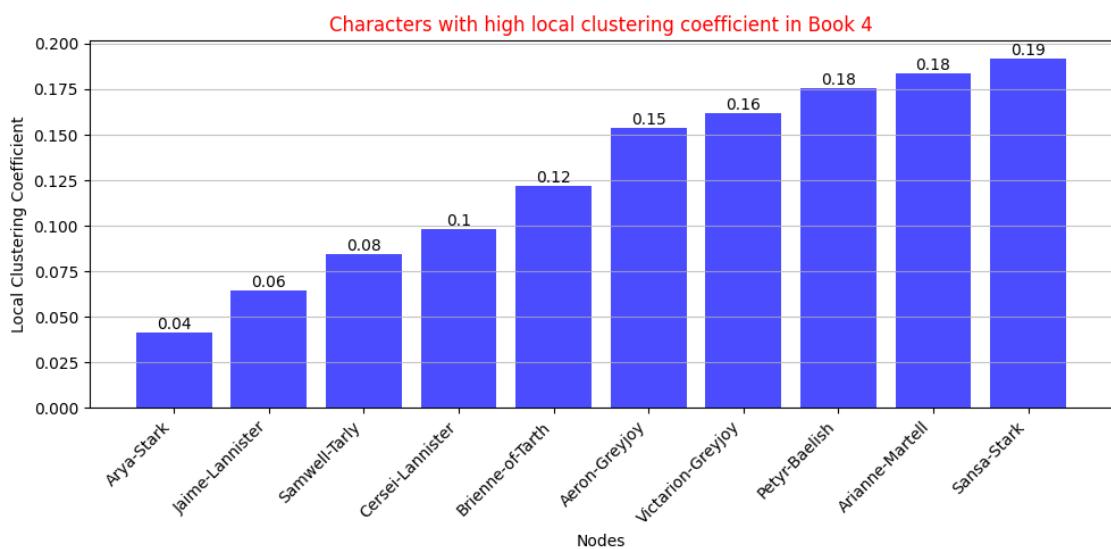


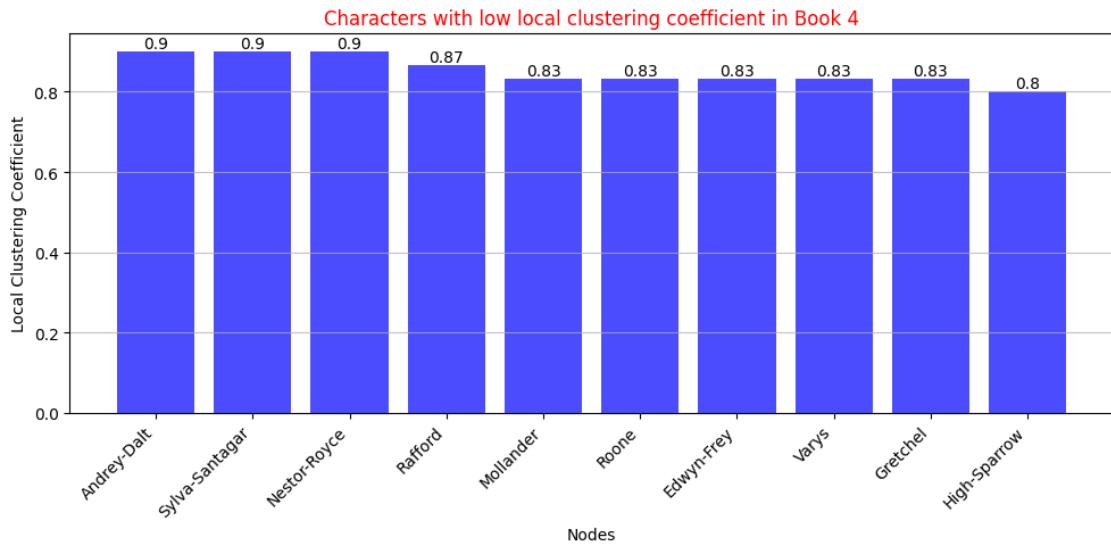
```
[ ]: plot_clustering_histogram(important_char_3, title="Characters with high local clustering coefficient in Book 3")
plot_clustering_histogram(not_important_char_3, title="Characters with low local clustering coefficient in Book 3")
```



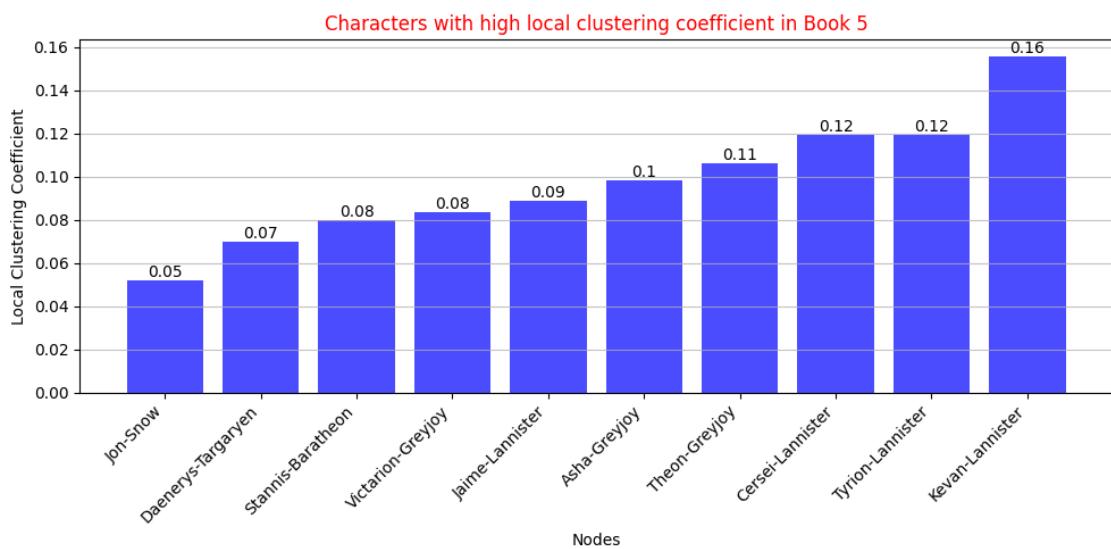


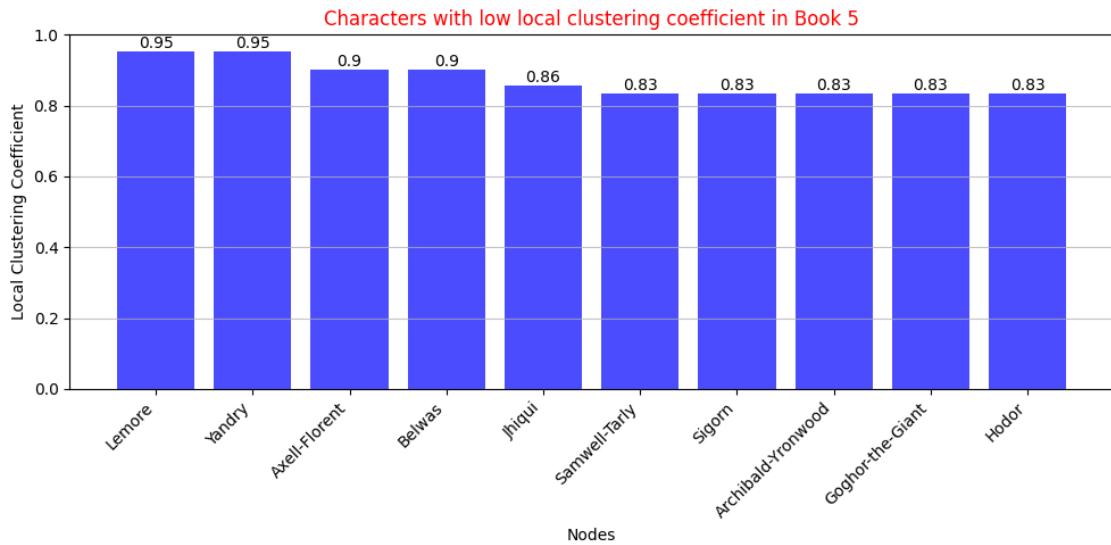
```
[ ]: plot_clustering_histogram(important_char_4, title="Characters with high local clustering coefficient in Book 4")
plot_clustering_histogram(not_important_char_4, title="Characters with low local clustering coefficient in Book 4")
```



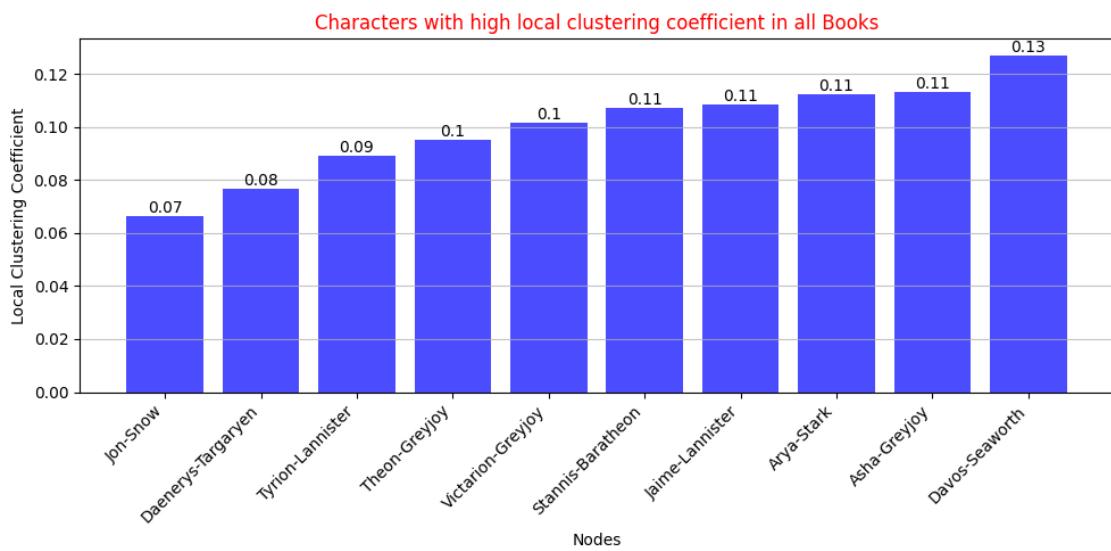


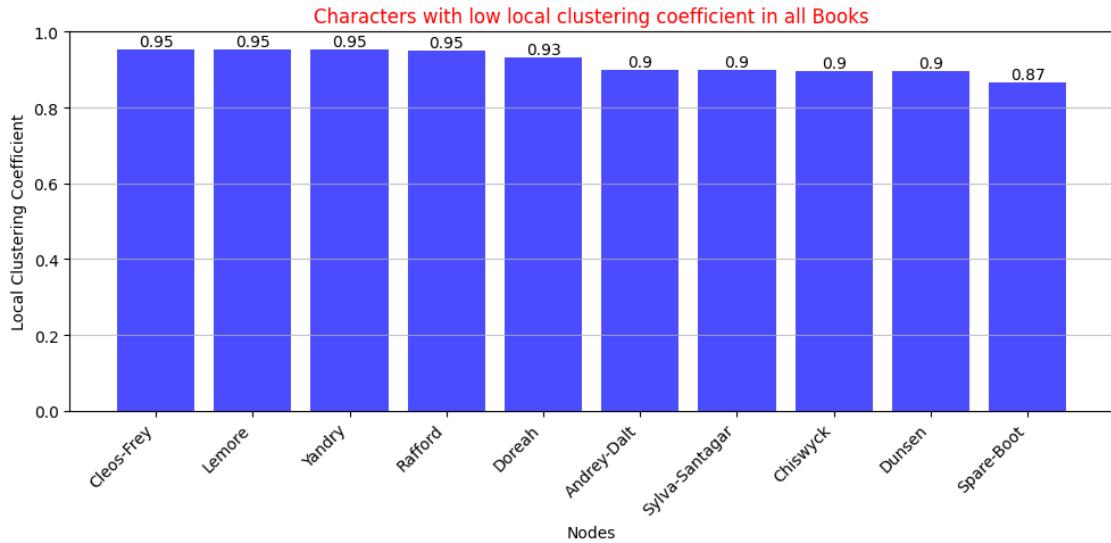
```
[ ]: plot_clustering_histogram(important_char_5, title="Characters with high local clustering coefficient in Book 5")
plot_clustering_histogram(not_important_char_5, title="Characters with low local clustering coefficient in Book 5")
```





```
[ ]: plot_clustering_histogram(important_char_all, title="Characters with high local clustering coefficient in all Books")
plot_clustering_histogram(not_important_char_all, title="Characters with low local clustering coefficient in all Books")
```





6.1.7 Similarity-Equivalence

Structural Equivalence

Jaccard Coefficient

```
[ ]: def top_bottom_n_jaccard_pairs(jaccard_coefficients, n):
    """
    Trova le n coppie di nodi con il coefficiente di Jaccard più alto e più
    basso.

    :param jaccard_coefficients: Generatore di coefficienti di Jaccard (u, v, ↴p).
    :param n: Numero di coppie da restituire per top e bottom.
    :return: Tuple con due liste: (top_pairs, bottom_pairs).
    """

    # Creare una lista per memorizzare le coppie e i loro coefficienti
    jaccard_list = [(p, (u, v)) for u, v, p in jaccard_coefficients]

    # Trovare le n coppie con il coefficiente di Jaccard più alto usando un heap
    top_pairs = heapq.nlargest(n, jaccard_list)

    # Trovare le n coppie con il coefficiente di Jaccard più basso usando un
    ↴heap
    bottom_pairs = heapq.nsmallest(n, jaccard_list)

    return [(pair, coeff) for coeff, pair in top_pairs], [(pair, coeff) for
    ↴coeff, pair in bottom_pairs]
```

```
[ ]: j_1 = nx.jaccard_coefficient(G_1, ebunch=None)
j_2 = nx.jaccard_coefficient(G_2, ebunch=None)
j_3 = nx.jaccard_coefficient(G_3, ebunch=None)
j_4 = nx.jaccard_coefficient(G_4, ebunch=None)
j_5 = nx.jaccard_coefficient(G_5, ebunch=None)
j_all = nx.jaccard_coefficient(G_all, ebunch=None)

[ ]: top_jaccard_pairs_1, bottom_jaccard_pairs_1 = top_bottom_n_jaccard_pairs(j_1, ↵15)
top_jaccard_pairs_2, bottom_jaccard_pairs_2 = top_bottom_n_jaccard_pairs(j_2, ↵15)
top_jaccard_pairs_3, bottom_jaccard_pairs_3 = top_bottom_n_jaccard_pairs(j_3, ↵15)
top_jaccard_pairs_4, bottom_jaccard_pairs_4 = top_bottom_n_jaccard_pairs(j_4, ↵15)
top_jaccard_pairs_5, bottom_jaccard_pairs_5 = top_bottom_n_jaccard_pairs(j_5, ↵15)
top_jaccard_pairs_all, bottom_jaccard_pairs_all = ↵top_bottom_n_jaccard_pairs(j_all, 15)

[ ]: print("Nodes with high Jaccard Coefficient in Book 1: ")
for el in top_jaccard_pairs_1:
    print(el)

print("\nNodes with high Jaccard Coefficient in Book 2: ")
for el in top_jaccard_pairs_2:
    print(el)

print("\nNodes with high Jaccard Coefficient in Book 3: ")
for el in top_jaccard_pairs_3:
    print(el)

print("\nNodes with high Jaccard Coefficient in Book 4: ")
for el in top_jaccard_pairs_4:
    print(el)

print("\nNodes with high Jaccard Coefficient in Book 5: ")
for el in top_jaccard_pairs_5:
    print(el)

print("\nNodes with high Jaccard Coefficient in all Books: ")
for el in top_jaccard_pairs_all:
    print(el)
```

Nodes with high Jaccard Coefficient in Book 1:
 (('Wylla', 'Varly'), 1.0)
 (('Wylla', 'Tregar'), 1.0)

```
(('Wylla', 'Tobho-Mott'), 1.0)
((('Wylla', 'Porther'), 1.0)
((('Wylla', 'Joss'), 1.0)
((('Wylla', 'Jacks'), 1.0)
((('Wylla', 'Howland-Reed'), 1.0)
((('Varly', 'Tregar'), 1.0)
((('Varly', 'Porther'), 1.0)
((('Varly', 'Jacks'), 1.0)
((('Tregar', 'Porther'), 1.0)
((('Tregar', 'Jacks'), 1.0)
((('Tobho-Mott', 'Varly'), 1.0)
((('Tobho-Mott', 'Tregar'), 1.0)
((('Tobho-Mott', 'Porther'), 1.0)
```

Nodes with high Jaccard Coefficient in Book 2:

```
((('Robert-Arryn', 'Trystane-Martell'), 1.0)
((('Rickard-Karstark', 'Martyn-Rivers'), 1.0)
((('Rickard-Karstark', 'Jon-Umber-(Greatjon)'), 1.0)
((('Quaith', 'Mirri-Maz-Duur'), 1.0)
((('Quaith', 'Lynesse-Hightower'), 1.0)
((('Quaith', 'Doreah'), 1.0)
((('Preston-Greenfield', 'Vylarr'), 1.0)
((('Praed', 'Dobber'), 1.0)
((('Praed', 'Aenys-Frey'), 1.0)
((('Olyvar-Frey', 'Rickard-Karstark'), 1.0)
((('Olyvar-Frey', 'Martyn-Rivers'), 1.0)
((('Olyvar-Frey', 'Jon-Umber-(Greatjon)'), 1.0)
((('Matthos-Seaworth', 'Imry-Florent'), 1.0)
((('Martyn-Rivers', 'Jon-Umber-(Greatjon)'), 1.0)
((('Lynesse-Hightower', 'Mirri-Maz-Duur'), 1.0)
```

Nodes with high Jaccard Coefficient in Book 3:

```
((('Yohn-Royce', 'Grisel'), 1.0)
((('Urswyck', 'Aemon-Targaryen-(Dragonknight)'), 1.0)
((('Tristifer-IV-Mudd', 'Rolph-Spicer'), 1.0)
((('Tristifer-IV-Mudd', 'Perwyn-Frey'), 1.0)
((('Rolph-Spicer', 'Perwyn-Frey'), 1.0)
((('Rodrik-Cassel', 'Brenett'), 1.0)
((('Ravella-Swann', 'Pate-(Old)'), 1.0)
((('Ravella-Swann', 'Mad-Huntsman'), 1.0)
((('Pate-(Old)', 'Mad-Huntsman'), 1.0)
((('Patchface', 'Daeron-I-Targaryen'), 1.0)
((('Ottyn-Wythers', 'Mallador-Locke'), 1.0)
((('Orell', 'Big-Boil'), 1.0)
((('Kella', 'Mordane'), 1.0)
((('Groleo', 'Rhaegel-Targaryen'), 1.0)
((('Galyeon-of-Cuy', 'Chataya'), 1.0)
```

Nodes with high Jaccard Coefficient in Book 4:

```
(('Yorko-Terys', 'Umma'), 1.0)
((('Wylis-Manderly', 'Ronnet-Connington'), 1.0)
((('Unella', 'Baelor-I-Targaryen'), 1.0)
((('Theobald', 'Qhorin-Halfhand'), 1.0)
((('Talea', 'Brusco'), 1.0)
((('Talbert-Serry', 'Ralf-(Limper)'), 1.0)
((('Talbert-Serry', 'Harras-Harlaw'), 1.0)
((('Syrio-Forel', 'Yorko-Terys'), 1.0)
((('Syrio-Forel', 'Umma'), 1.0)
((('Syrio-Forel', 'Meralyn'), 1.0)
((('Syrio-Forel', 'Bellegere-Otherys'), 1.0)
((('Sybell-Spicer', 'Wylis-Manderly'), 1.0)
((('Sybell-Spicer', 'Ronnet-Connington'), 1.0)
((('Sybell-Spicer', 'Hugo-Vance'), 1.0)
((('Sybell-Spicer', 'Garrett-Paege'), 1.0)
```

Nodes with high Jaccard Coefficient in Book 5:

```
((('Yurkhaz-zo-Yunzak', 'Aegon-I-Targaryen'), 1.0)
((('Ygritte', 'Rattleshirt'), 1.0)
((('Wulfe', 'Ralf-(Limper)'), 1.0)
((('Wulfe', 'Longwater-Pyke'), 1.0)
((('Wulfe', 'Kerwin'), 1.0)
((('William-Foxglove', 'Roose-Ryswell'), 1.0)
((('Widower', 'Larraq'), 1.0)
((('Wex-Pyke', 'Willow-Witch-eye'), 1.0)
((('Ulmer', 'Soren-Shieldbreaker'), 1.0)
((('Ulmer', 'Benjen-Stark'), 1.0)
((('Ulmer', 'Alliser-Thorne'), 1.0)
((('Tytos-Blackwood', 'Jonos-Bracken'), 1.0)
((('Tytos-Blackwood', 'Hoster-Blackwood'), 1.0)
((('Tytos-Blackwood', 'Brynden-Tully'), 1.0)
((('Torgon-Greyiron', 'Galbart-Glover'), 1.0)
```

Nodes with high Jaccard Coefficient in all Books:

```
((('Zei', 'Ulmer'), 1.0)
((('Zei', 'Soren-Shieldbreaker'), 1.0)
((('Zei', 'Othor'), 1.0)
((('Zei', 'Matthar'), 1.0)
((('Zei', 'Hake'), 1.0)
((('Zei', 'Arson'), 1.0)
((('Yurkhaz-zo-Yunzak', 'Rhaegel-Targaryen'), 1.0)
((('Yorko-Terys', 'Umma'), 1.0)
((('Yorko-Terys', 'Ravella-Swann'), 1.0)
((('Yorko-Terys', 'Praed'), 1.0)
((('Yorko-Terys', 'Mad-Huntsman'), 1.0)
((('Yorko-Terys', 'Dobber'), 1.0)
((('Yorko-Terys', 'Bellegere-Otherys'), 1.0)
```

```
(('Wynton-Stout', 'Maekar-I-Targaryen'), 1.0)
((('Wynton-Stout', 'Aegon-V-Targaryen'), 1.0)
```

Sono praticamente tutte comparse. Ha senso che siano strutturalmente simili perché si tratta di nodi che tendono ad essere debolmente collegati ad altri nodi, cioè tendono ad essere collegati ai personaggi principali.

```
[ ]: print("Nodes with low Jaccard Coefficient in Book 1: ")
for el in bottom_jaccard_pairs_1:
    print(el)

print("\nNodes with low Jaccard Coefficient in Book 2: ")
for el in bottom_jaccard_pairs_2:
    print(el)

print("\nNodes with low Jaccard Coefficient in Book 3: ")
for el in bottom_jaccard_pairs_3:
    print(el)

print("\nNodes with low Jaccard Coefficient in Book 4: ")
for el in bottom_jaccard_pairs_4:
    print(el)

print("\nNodes with low Jaccard Coefficient in Book 5: ")
for el in bottom_jaccard_pairs_5:
    print(el)

print("\nNodes with low Jaccard Coefficient in all Books: ")
for el in bottom_jaccard_pairs_all:
    print(el)
```

```
Nodes with low Jaccard Coefficient in Book 1:
((('Addam-Marbrand', 'Balon-Greyjoy'), 0.0)
((('Addam-Marbrand', 'Brandon-Stark'), 0.0)
((('Addam-Marbrand', 'Chett'), 0.0)
((('Addam-Marbrand', 'Clement-Piper'), 0.0)
((('Addam-Marbrand', 'Cohollo'), 0.0)
((('Addam-Marbrand', 'Conn'), 0.0)
((('Addam-Marbrand', 'Desmond'), 0.0)
((('Addam-Marbrand', 'Donnel-Waynwood'), 0.0)
((('Addam-Marbrand', 'Eroeh'), 0.0)
((('Addam-Marbrand', 'Gared'), 0.0)
((('Addam-Marbrand', 'Gunthor-son-of-Gurn'), 0.0)
((('Addam-Marbrand', 'Hali'), 0.0)
((('Addam-Marbrand', 'Harwin'), 0.0)
((('Addam-Marbrand', 'Heward'), 0.0)
((('Addam-Marbrand', 'Jacks'), 0.0)
```

Nodes with low Jaccard Coefficient in Book 2:

```
(('Aegon-I-Targaryen', 'Aenys-Frey'), 0.0)
((('Aegon-I-Targaryen', 'Aeron-Greyjoy'), 0.0)
((('Aegon-I-Targaryen', 'Aggo'), 0.0)
((('Aegon-I-Targaryen', 'Alester-Florent'), 0.0)
((('Aegon-I-Targaryen', 'Amabel'), 0.0)
((('Aegon-I-Targaryen', 'Ardrian-Celtigar'), 0.0)
((('Aegon-I-Targaryen', 'Aron-Santagar'), 0.0)
((('Aegon-I-Targaryen', 'Axell-Florent'), 0.0)
((('Aegon-I-Targaryen', 'Balon-Greyjoy'), 0.0)
((('Aegon-I-Targaryen', 'Chayle'), 0.0)
((('Aegon-I-Targaryen', 'Chett'), 0.0)
((('Aegon-I-Targaryen', 'Chiswyck'), 0.0)
((('Aegon-I-Targaryen', 'Cley-Cerwyn'), 0.0)
((('Aegon-I-Targaryen', 'Colen-of-Greenpools'), 0.0)
((('Aegon-I-Targaryen', 'Cutjack'), 0.0)
```

Nodes with low Jaccard Coefficient in Book 3:

```
((('Addam-Marbrand', 'Aegon-Frey-(son-of-Stevron)'), 0.0)
((('Addam-Marbrand', 'Aegon-Targaryen-(son-of-Rhaegar)'), 0.0)
((('Addam-Marbrand', 'Alester-Florent'), 0.0)
((('Addam-Marbrand', 'Alla-Tyrell'), 0.0)
((('Addam-Marbrand', 'Balon-Greyjoy'), 0.0)
((('Addam-Marbrand', 'Bedwyck'), 0.0)
((('Addam-Marbrand', 'Big-Boil'), 0.0)
((('Addam-Marbrand', 'Brandon-Stark'), 0.0)
((('Addam-Marbrand', 'Brenett'), 0.0)
((('Addam-Marbrand', 'Chett'), 0.0)
((('Addam-Marbrand', 'Clubfoot-Karl'), 0.0)
((('Addam-Marbrand', 'Cotter-Pyke'), 0.0)
((('Addam-Marbrand', 'Daeron-I-Targaryen'), 0.0)
((('Addam-Marbrand', 'Dalla'), 0.0)
((('Addam-Marbrand', 'Denys-Mallister'), 0.0)
```

Nodes with low Jaccard Coefficient in Book 4:

```
((('Addam-Marbrand', 'Aegon-Targaryen-(son-of-Rhaegar)'), 0.0)
((('Addam-Marbrand', 'Aeron-Greyjoy'), 0.0)
((('Addam-Marbrand', 'Alla-Tyrell'), 0.0)
((('Addam-Marbrand', 'Alleras'), 0.0)
((('Addam-Marbrand', 'Alys-Arryn'), 0.0)
((('Addam-Marbrand', 'Balon-Greyjoy'), 0.0)
((('Addam-Marbrand', 'Bellegere-Otherys'), 0.0)
((('Addam-Marbrand', 'Benedar-Belmore'), 0.0)
((('Addam-Marbrand', 'Clarence-Crabb'), 0.0)
((('Addam-Marbrand', 'Daemon-Sand'), 0.0)
((('Addam-Marbrand', 'Dalla'), 0.0)
((('Addam-Marbrand', 'Denyo-Terys'), 0.0)
((('Addam-Marbrand', 'Elys-Waynwood'), 0.0)
```

```
(('Addam-Marbrand', 'Gormon-Tyrell'), 0.0)
((('Addam-Marbrand', 'Gyles-Grafton'), 0.0)

Nodes with low Jaccard Coefficient in Book 5:
((('Aegon-I-Targaryen', 'Aenys-Frey'), 0.0)
((('Aegon-I-Targaryen', 'Balon-Greyjoy'), 0.0)
((('Aegon-I-Targaryen', 'Benethon-Scales'), 0.0)
((('Aegon-I-Targaryen', 'Camarron'), 0.0)
((('Aegon-I-Targaryen', 'Denys-Mallister'), 0.0)
((('Aegon-I-Targaryen', 'Devan-Seaworth'), 0.0)
((('Aegon-I-Targaryen', 'Fulk'), 0.0)
((('Aegon-I-Targaryen', 'High-Sparrow'), 0.0)
((('Aegon-I-Targaryen', 'Hodor'), 0.0)
((('Aegon-I-Targaryen', 'Jared-Frey'), 0.0)
((('Aegon-I-Targaryen', 'Jeyne-Poole'), 0.0)
((('Aegon-I-Targaryen', 'Kasporio'), 0.0)
((('Aegon-I-Targaryen', 'Leaf'), 0.0)
((('Aegon-I-Targaryen', 'Longwater-Pyke'), 0.0)
((('Aegon-I-Targaryen', 'Marlon-Manderly'), 0.0)
```

```
Nodes with low Jaccard Coefficient in all Books:
((('Addam-Marbrand', 'Aenys-Frey'), 0.0)
((('Addam-Marbrand', 'Alla-Tyrell'), 0.0)
((('Addam-Marbrand', 'Aron-Santagar'), 0.0)
((('Addam-Marbrand', 'Bellegere-Otherys'), 0.0)
((('Addam-Marbrand', 'Benethon-Scales'), 0.0)
((('Addam-Marbrand', 'Camarron'), 0.0)
((('Addam-Marbrand', 'Clubfoot-Karl'), 0.0)
((('Addam-Marbrand', 'Denyo-Terys'), 0.0)
((('Addam-Marbrand', 'Denys-Mallister'), 0.0)
((('Addam-Marbrand', 'Devan-Seaworth'), 0.0)
((('Addam-Marbrand', 'Frenken'), 0.0)
((('Addam-Marbrand', 'Fulk'), 0.0)
((('Addam-Marbrand', 'Galazza-Galare'), 0.0)
((('Addam-Marbrand', 'Gormon-Tyrell'), 0.0)
((('Addam-Marbrand', 'Gyles-Grafton'), 0.0)
```

Pearson Correlation Coefficient

```
[ ]: def pearson_correlation_coefficient(graph):
    # Ottieni i nomi dei nodi
    node_names = list(graph.nodes())
    n = len(node_names)

    # Inizializza una lista per i risultati
    correlation_results = []

    # Calcola la matrice di adiacenza
```

```

A = nx.adjacency_matrix(graph).todense() # Usa .todense() per ottenere un array NumPy

# Calcola il coefficiente di correlazione di Pearson per ogni coppia di nodi
for i in range(n):
    for j in range(i + 1, n): # Solo coppie uniche (i < j)
        # Estrai i vettori di adiacenza per i nodi i e j
        vec_i = A[i, :] # Converti la riga in un array 1D
        vec_j = A[j, :]

        # Calcola il coefficiente di correlazione di Pearson
        if np.std(vec_i) > 0 and np.std(vec_j) > 0: # Evita divisione per zero
            correlation = np.corrcoef(vec_i, vec_j)[0, 1]
            correlation_results.append((node_names[i], node_names[j], correlation))
        else:
            correlation_results.append((node_names[i], node_names[j], None)) # Nessuna correlazione

return correlation_results

```

```

[ ]: def bottom_top_n_similar_pairs(similarity_results, n):
    # Ordina i risultati in base al punteggio di similarità in ordine decrescente
    sorted_results = sorted(similarity_results, key=lambda x: x[2], reverse=True)

    # Prendi le prime n coppie
    top_pairs = sorted_results[:n]
    bottom_pairs = sorted_results[-n:]

    return bottom_pairs, top_pairs

```

```

[ ]: pearson_corr_1 = pearson_correlation_coefficient(G_1)
pearson_corr_2 = pearson_correlation_coefficient(G_2)
pearson_corr_3 = pearson_correlation_coefficient(G_3)
pearson_corr_4 = pearson_correlation_coefficient(G_4)
pearson_corr_5 = pearson_correlation_coefficient(G_5)
pearson_corr_all = pearson_correlation_coefficient(G_all)

```

```

[ ]: bottom_pearson_corr_1, top_pearson_corr_1 = bottom_top_n_similar_pairs(pearson_corr_1, 15)
bottom_pearson_corr_2, top_pearson_corr_2 = bottom_top_n_similar_pairs(pearson_corr_2, 15)

```

```

bottom_pearson_corr_3, top_pearson_corr_3 = ↵
    ↵bottom_top_n_similar_pairs(pearson_corr_3, 15)
bottom_pearson_corr_4, top_pearson_corr_4 = ↵
    ↵bottom_top_n_similar_pairs(pearson_corr_4, 15)
bottom_pearson_corr_5, top_pearson_corr_5 = ↵
    ↵bottom_top_n_similar_pairs(pearson_corr_4, 15)
bottom_pearson_corr_all, top_pearson_corr_all = ↵
    ↵bottom_top_n_similar_pairs(pearson_corr_all, 15)

```

```

[ ]: print("Nodes with high Pearson Correlation Coefficient in Book 1: ")
for el in top_pearson_corr_1:
    print(el)

print("\nNodes with high Pearson Correlation Coefficient in Book 2: ")
for el in top_pearson_corr_2:
    print(el)

print("\nNodes with high Pearson Correlation Coefficient in Book 3: ")
for el in top_pearson_corr_3:
    print(el)

print("\nNodes with high Pearson Correlation Coefficient in Book 4: ")
for el in top_pearson_corr_4:
    print(el)

print("\nNodes with high Pearson Correlation Coefficient in Book 5: ")
for el in top_pearson_corr_5:
    print(el)

print("\nNodes with high Pearson Correlation Coefficient in all Books: ")
for el in top_pearson_corr_all:
    print(el)

```

Nodes with high Pearson Correlation Coefficient in Book 1:

```

('Donnel-Waynwood', 'Moreo-Tumitis', np.float64(1.0))
('Donnel-Waynwood', 'Mychel-Redfort', np.float64(1.0))
('Masha-Heddle', 'Mya-Stone', np.float64(1.0))
('Moreo-Tumitis', 'Mychel-Redfort', np.float64(1.0))
('Chella', 'Kurleket', np.float64(1.0))
('Chella', 'Leo-Lefford', np.float64(1.0))
('Chella', 'Mord', np.float64(1.0))
('Dareon', 'Matthar', np.float64(1.0))
('Dolf', 'Ulf-son-of-Umar', np.float64(1.0))
('Donal-Noye', 'Dywen', np.float64(1.0))
('Dywen', 'Hobb', np.float64(1.0))
('Dywen', 'Othor', np.float64(1.0))
('Jacks', 'Joss', np.float64(1.0))

```

```

('Jacks', 'Tobho-Mott', np.float64(1.0))
('Joss', 'Porther', np.float64(1.0))

Nodes with high Pearson Correlation Coefficient in Book 2:
('Aegon-V-Targaryen', 'Daeron-II-Targaryen', np.float64(1.0))
('Aegon-V-Targaryen', 'Maekar-I-Targaryen', np.float64(1.0))
('Daeron-II-Targaryen', 'Maekar-I-Targaryen', np.float64(1.0))
('Aenys-Frey', 'Praed', np.float64(1.0))
('Howland-Reed', 'Poxy-Tym', np.float64(1.0))
('Captain-Myraham', 'Euron-Greyjoy', np.float64(1.0))
('Captain-Myraham', 'Jason-Mallister', np.float64(1.0))
('Captain-Myraham', 'Kyra', np.float64(1.0))
('Captain-Myraham-Daughter', 'Farlen', np.float64(1.0))
('Chett', 'Hake', np.float64(1.0))
('Chett', 'Lark', np.float64(1.0))
('Doreah', 'Lynesse-Hightower', np.float64(1.0))
('Doreah', 'Mirri-Maz-Duur', np.float64(1.0))
('Lynesse-Hightower', 'Mirri-Maz-Duur', np.float64(1.0))
('Euron-Greyjoy', 'Jason-Mallister', np.float64(1.0))

Nodes with high Pearson Correlation Coefficient in Book 3:
('Aemon-Targaryen-(Dragonknight)', 'Urswyck', np.float64(1.0))
('Mad-Huntsman', 'Pate-(Old)', np.float64(1.0))
('Mad-Huntsman', 'Ravella-Swann', np.float64(1.0))
('Pate-(Old)', 'Ravella-Swann', np.float64(1.0))
('Brenett', 'Rodrik-Cassel', np.float64(1.0))
('Daeron-I-Targaryen', 'Patchface', np.float64(1.0))
('Perwyn-Frey', 'Tristifer-IV-Mudd', np.float64(1.0))
('Arson', 'Zei', np.float64(0.9999999999999999))
('Grisel', 'Yohn-Royce', np.float64(0.9999999999999999))
('Mallador-Locke', 'Ottyn-Wythers', np.float64(0.9999999999999992))
('Kella', 'Mordane', np.float64(0.999999999999984))
('Grolio', 'Rhaegel-Targaryen', np.float64(0.999999999999982))
('Perwyn-Frey', 'Rolph-Spicer', np.float64(0.999999999999982))
('Rolph-Spicer', 'Tristifer-IV-Mudd', np.float64(0.999999999999982))
('Chataya', 'Galyeon-of-Cuy', np.float64(0.999999999999974))

Nodes with high Pearson Correlation Coefficient in Book 4:
('Emmond', 'Gormond-Goodbrother', np.float64(1.0))
('Emmond', 'Greydon-Goodbrother', np.float64(1.0))
('Emmond', 'Rus', np.float64(1.0))
('Gormond-Goodbrother', 'Greydon-Goodbrother', np.float64(1.0))
('Gormond-Goodbrother', 'Rus', np.float64(1.0))
('Greydon-Goodbrother', 'Rus', np.float64(1.0))
('Barristan-Selmy', 'Denys-Darklyn', np.float64(1.0))
('Barristan-Selmy', 'Elbert-Arryn', np.float64(1.0))
('Barristan-Selmy', 'Owen-Merryweather', np.float64(1.0))
('Denys-Darklyn', 'Elbert-Arryn', np.float64(1.0))

```

```

('Denys-Darklyn', 'Owen-Merryweather', np.float64(1.0))
('Franklyn-Fowler', 'Timoth', np.float64(1.0))
('Bellegere-Otherys', 'Syrio-Forel', np.float64(1.0))
('Bellegere-Otherys', 'Umma', np.float64(1.0))
('Bellegere-Otherys', 'Yorko-Terys', np.float64(1.0))

Nodes with high Pearson Correlation Coefficient in Book 5:
('Emmond', 'Gormond-Goodbrother', np.float64(1.0))
('Emmond', 'Greydon-Goodbrother', np.float64(1.0))
('Emmond', 'Rus', np.float64(1.0))
('Gormond-Goodbrother', 'Greydon-Goodbrother', np.float64(1.0))
('Gormond-Goodbrother', 'Rus', np.float64(1.0))
('Greydon-Goodbrother', 'Rus', np.float64(1.0))
('Barristan-Selmy', 'Denys-Darklyn', np.float64(1.0))
('Barristan-Selmy', 'Elbert-Arryn', np.float64(1.0))
('Barristan-Selmy', 'Owen-Merryweather', np.float64(1.0))
('Denys-Darklyn', 'Elbert-Arryn', np.float64(1.0))
('Denys-Darklyn', 'Owen-Merryweather', np.float64(1.0))
('Franklyn-Fowler', 'Timoth', np.float64(1.0))
('Bellegere-Otherys', 'Syrio-Forel', np.float64(1.0))
('Bellegere-Otherys', 'Umma', np.float64(1.0))
('Bellegere-Otherys', 'Yorko-Terys', np.float64(1.0))

```

```

Nodes with high Pearson Correlation Coefficient in all Books:
('Donnel-Waynwood', 'Masha-Heddle', np.float64(1.0))
('Masha-Heddle', 'Moreo-Tumitis', np.float64(1.0))
('Masha-Heddle', 'Mychel-Redfort', np.float64(1.0))
('Masha-Heddle', 'Brenett', np.float64(1.0))
('Eroeh', 'Lynesse-Hightower', np.float64(1.0))
('Eroeh', 'Quaithe', np.float64(1.0))
('Eroeh', 'Rhaegel-Targaryen', np.float64(1.0))
('Eroeh', 'Ezzara', np.float64(1.0))
('Eroeh', 'Lysono-Maar', np.float64(1.0))
('Jacks', 'Joss', np.float64(1.0))
('Jacks', 'Porther', np.float64(1.0))
('Jacks', 'Tobho-Mott', np.float64(1.0))
('Jacks', 'Tregar', np.float64(1.0))
('Jacks', 'Varly', np.float64(1.0))
('Joss', 'Porther', np.float64(1.0))

```

```

[ ]: print("Nodes with low Pearson Correlation Coefficient in Book 1: ")
for el in bottom_pearson_corr_1:
    print(el)

print("\nNodes with low Pearson Correlation Coefficient in Book 2: ")
for el in bottom_pearson_corr_2:
    print(el)

```

```

print("\nNodes with low Pearson Correlation Coefficient in Book 3: ")
for el in bottom_pearson_corr_3:
    print(el)

print("\nNodes with low Pearson Correlation Coefficient in Book 4: ")
for el in bottom_pearson_corr_4:
    print(el)

print("\nNodes with low Pearson Correlation Coefficient in Book 5: ")
for el in bottom_pearson_corr_5:
    print(el)

print("\nNodes with low Pearson Correlation Coefficient in all Books: ")
for el in bottom_pearson_corr_all:
    print(el)

```

Nodes with low Pearson Correlation Coefficient in Book 1:

('Jon-Snow', 'Irri', np.float64(-0.056819680484388194))
 ('Jhogo', 'Tyrion-Lannister', np.float64(-0.05701058476448387))
 ('Tyrion-Lannister', 'Doreah', np.float64(-0.058475727237995774))
 ('Jon-Snow', 'Qotho', np.float64(-0.05854715120216747))
 ('Daenerys-Targaryen', 'Jon-Snow', np.float64(-0.05865064625205858))
 ('Catelyn-Stark', 'Irri', np.float64(-0.05896097779180299))
 ('Rakharo', 'Tyrion-Lannister', np.float64(-0.0593141652510554))
 ('Quaro', 'Tyrion-Lannister', np.float64(-0.05978250233436808))
 ('Catelyn-Stark', 'Qotho', np.float64(-0.06075354969222039))
 ('Drogo', 'Tyrion-Lannister', np.float64(-0.061166552559389055))
 ('Tyrion-Lannister', 'Haggo', np.float64(-0.06148442995251337))
 ('Aggo', 'Tyrion-Lannister', np.float64(-0.06250291739312593))
 ('Tyrion-Lannister', 'Irri', np.float64(-0.06511124860069602))
 ('Tyrion-Lannister', 'Qotho', np.float64(-0.06709080523313271))
 ('Daenerys-Targaryen', 'Tyrion-Lannister', np.float64(-0.06741264266424296))

Nodes with low Pearson Correlation Coefficient in Book 2:

('Theon-Greyjoy', 'Gyles-Rosby', np.float64(-0.05037961315578024))
 ('Daenerys-Targaryen', 'Amory-Lorch', np.float64(-0.05306846890095701))
 ('Theon-Greyjoy', 'Sandor-Clegane', np.float64(-0.054500466860532976))
 ('Theon-Greyjoy', 'Meryn-Trant', np.float64(-0.05621023302888922))
 ('Theon-Greyjoy', 'Daenerys-Targaryen', np.float64(-0.05622822709630116))
 ('Arya-Stark', 'Daenerys-Targaryen', np.float64(-0.0563351063867637))
 ('Theon-Greyjoy', 'Tickler', np.float64(-0.05637558926682269))
 ('Tyrion-Lannister', 'Daenerys-Targaryen', np.float64(-0.05651209060940125))
 ('Theon-Greyjoy', 'Rafford', np.float64(-0.05663410309792912))
 ('Theon-Greyjoy', 'Chiswyck', np.float64(-0.058253901577104736))
 ('Theon-Greyjoy', 'Polliver', np.float64(-0.05825743388419749))
 ('Theon-Greyjoy', 'Dunsen', np.float64(-0.05879642160269929))

```
('Daenerys-Targaryen', 'Robb-Stark', np.float64(-0.06095698437281538))
('Theon-Greyjoy', 'Gregor-Clegane', np.float64(-0.06320908658441911))
('Theon-Greyjoy', 'Amory-Lorch', np.float64(-0.06980426519231432))
```

Nodes with low Pearson Correlation Coefficient in Book 3:

```
('Samwell-Tarly', 'Meryn-Trant', np.float64(-0.04982916829740947))
('Tyrion-Lannister', 'Samwell-Tarly', np.float64(-0.05028118123849419))
('Daenerys-Targaryen', 'Samwell-Tarly', np.float64(-0.05169527211885869))
('Samwell-Tarly', 'Sansa-Stark', np.float64(-0.051837371518053524))
('Jon-Snow', 'Arya-Stark', np.float64(-0.052097405952195856))
('Tyrion-Lannister', 'Daenerys-Targaryen', np.float64(-0.05215814042356369))
('Daenerys-Targaryen', 'Sansa-Stark', np.float64(-0.05254242510749202))
('Daenerys-Targaryen', 'Arya-Stark', np.float64(-0.0528243510828587))
('Tyrion-Lannister', 'Jeor-Mormont', np.float64(-0.052949565532283346))
('Robb-Stark', 'Eddison-Tollett', np.float64(-0.05362079084114234))
('Jaime-Lannister', 'Samwell-Tarly', np.float64(-0.054945157557264424))
('Samwell-Tarly', 'Cersei-Lannister', np.float64(-0.056706133759633924))
('Catelyn-Stark', 'Daenerys-Targaryen', np.float64(-0.056842289737753))
('Daenerys-Targaryen', 'Jon-Snow', np.float64(-0.05994350157152525))
('Robb-Stark', 'Daenerys-Targaryen', np.float64(-0.06832745464491957))
```

Nodes with low Pearson Correlation Coefficient in Book 4:

```
('Jaime-Lannister', 'Pate-(novice)', np.float64(-0.0578534392792448))
('Jaime-Lannister', 'Robert-Arryn', np.float64(-0.05811968567668195))
('Jaime-Lannister', 'Rodrik-Harlaw', np.float64(-0.0583172512672932))
('Jaime-Lannister', 'Garin-(orphan)', np.float64(-0.058698745922805266))
('Jaime-Lannister', 'Anya-Waynwood', np.float64(-0.05948421080212302))
('Cersei-Lannister', 'Aeron-Greyjoy', np.float64(-0.060534511356142684))
('Cersei-Lannister', 'Arianne-Martell', np.float64(-0.06078939875403278))
('Jaime-Lannister', 'Arya-Stark', np.float64(-0.06208786645679208))
('Cersei-Lannister', 'Arya-Stark', np.float64(-0.0629089936504195))
('Jaime-Lannister', 'Victarion-Greyjoy', np.float64(-0.06315201161092893))
('Jaime-Lannister', 'Alleras', np.float64(-0.0635247718735091))
('Jaime-Lannister', 'Tyene-Sand', np.float64(-0.06430678469249226))
('Jaime-Lannister', 'Samwell-Tarly', np.float64(-0.07209158533303434))
('Jaime-Lannister', 'Aeron-Greyjoy', np.float64(-0.07250035842549626))
('Jaime-Lannister', 'Arianne-Martell', np.float64(-0.08198643473497526))
```

Nodes with low Pearson Correlation Coefficient in Book 5:

```
('Jaime-Lannister', 'Pate-(novice)', np.float64(-0.0578534392792448))
('Jaime-Lannister', 'Robert-Arryn', np.float64(-0.05811968567668195))
('Jaime-Lannister', 'Rodrik-Harlaw', np.float64(-0.0583172512672932))
('Jaime-Lannister', 'Garin-(orphan)', np.float64(-0.058698745922805266))
('Jaime-Lannister', 'Anya-Waynwood', np.float64(-0.05948421080212302))
('Cersei-Lannister', 'Aeron-Greyjoy', np.float64(-0.060534511356142684))
('Cersei-Lannister', 'Arianne-Martell', np.float64(-0.06078939875403278))
('Jaime-Lannister', 'Arya-Stark', np.float64(-0.06208786645679208))
('Cersei-Lannister', 'Arya-Stark', np.float64(-0.0629089936504195))
```

```
('Jaime-Lannister', 'Victarion-Greyjoy', np.float64(-0.06315201161092893))
('Jaime-Lannister', 'Alleras', np.float64(-0.0635247718735091))
('Jaime-Lannister', 'Tyene-Sand', np.float64(-0.06430678469249226))
('Jaime-Lannister', 'Samwell-Tarly', np.float64(-0.07209158533303434))
('Jaime-Lannister', 'Aeron-Greyjoy', np.float64(-0.07250035842549626))
('Jaime-Lannister', 'Arianne-Martell', np.float64(-0.08198643473497526))
```

Nodes with low Pearson Correlation Coefficient in all Books:

```
('Eddard-Stark', 'Arianne-Martell', np.float64(-0.03195240655442412))
('Joffrey-Baratheon', 'Euron-Greyjoy', np.float64(-0.031970385103455325))
('Drogo', 'Sansa-Stark', np.float64(-0.032540695997100445))
('Samwell-Tarly', 'Sandor-Clegane', np.float64(-0.03269584008042262))
('Cersei-Lannister', 'Euron-Greyjoy', np.float64(-0.03296595565402362))
('Jon-Snow', 'Drogo', np.float64(-0.03324031869642428))
('Jon-Snow', 'Arianne-Martell', np.float64(-0.033243846041425565))
('Jaime-Lannister', 'Samwell-Tarly', np.float64(-0.03352303240894928))
('Daenerys-Targaryen', 'Catelyn-Stark', np.float64(-0.03458363539509524))
('Catelyn-Stark', 'Arianne-Martell', np.float64(-0.034711152120664744))
('Daenerys-Targaryen', 'Arya-Stark', np.float64(-0.03487129996043174))
('Sansa-Stark', 'Victarion-Greyjoy', np.float64(-0.036268879074897994))
('Joffrey-Baratheon', 'Victarion-Greyjoy', np.float64(-0.03645246958735634))
('Jaime-Lannister', 'Victarion-Greyjoy', np.float64(-0.037087718789793556))
('Jaime-Lannister', 'Drogo', np.float64(-0.03747974734361998))
```

I nodi che hanno Pearson Correlation Coefficient alto sono tutte comparse.

Quelli che hanno valori bassi sono interessanti perché rappresentano personaggi principali o secondari, non comparse. Il valore di Pearson basso è dovuto dal fatto che operano tutti in zone molto distanti tra loro. Quindi, sono personaggi tutti importanti ma che hanno vicini completamente diversi.

Hamming Distance

```
[ ]: def hamming_distance(graph):
    # Ottieni i nomi dei nodi
    node_names = list(graph.nodes())
    n = len(node_names)

    # Calcola la matrice di adiacenza
    A = nx.adjacency_matrix(graph).todense()  # Usa .todense() per ottenere un
                                                ↪array NumPy

    # Inizializza una lista per i risultati
    distance_results = []

    # Calcola la distanza di Hamming per ogni coppia di nodi
    for i in range(n):
        for j in range(i + 1, n):  # Solo coppie uniche (i < j)
            # Estrai i vettori di adiacenza per i nodi i e j
```

```

    vec_i = np.array(A[i, :]).flatten() # Converti la riga in un array
    ↵1D

    vec_j = np.array(A[j, :]).flatten()

    # Calcola la distanza di Hamming usando un ciclo for
    hamming_dist = 0
    for k in range(len(vec_i)):
        hamming_dist += (vec_i[k] - vec_j[k]) ** 2 # Somma delle
    ↵differenze al quadrato

    distance_results.append((node_names[i], node_names[j], ↵
    ↵hamming_dist))

    return distance_results

```

```
[ ]: hamming_1 = hamming_distance(G_1)
hamming_2 = hamming_distance(G_2)
hamming_3 = hamming_distance(G_3)
hamming_4 = hamming_distance(G_4)
hamming_5 = hamming_distance(G_5)
hamming_all = hamming_distance(G_all)
```

```
[ ]: bottom_hamming_1, top_hamming_1 = bottom_top_n_similar_pairs(hamming_1, 15)
bottom_hamming_2, top_hamming_2 = bottom_top_n_similar_pairs(hamming_2, 15)
bottom_hamming_3, top_hamming_3 = bottom_top_n_similar_pairs(hamming_3, 15)
bottom_hamming_4, top_hamming_4 = bottom_top_n_similar_pairs(hamming_4, 15)
bottom_hamming_5, top_hamming_5 = bottom_top_n_similar_pairs(hamming_5, 15)
bottom_hamming_all, top_hamming_all = bottom_top_n_similar_pairs(hamming_all, ↵
    ↵15)
```

```
[ ]: print("Nodes with high Hamming Distance in Book 1: ")
for el in top_hamming_1:
    print(el)

print("\nNodes with high Hamming Distance in Book 2: ")
for el in top_hamming_2:
    print(el)

print("\nNodes with high Hamming Distance in Book 3: ")
for el in top_hamming_3:
    print(el)

print("\nNodes with high Hamming Distance in Book 4: ")
for el in top_hamming_4:
    print(el)

print("\nNodes with high Hamming Distance in Book 5: ")
```

```

for el in top_hamming_5:
    print(el)

print("\nNodes with high Hamming Distance in all Books: ")
for el in top_hamming_all:
    print(el)

```

Nodes with high Hamming Distance in Book 1:

```

('Eddard-Stark', 'Robert-Baratheon', np.int64(178485))
('Daenerys-Targaryen', 'Eddard-Stark', np.int64(133629))
('Eddard-Stark', 'Jon-Snow', np.int64(130238))
('Eddard-Stark', 'Drogo', np.int64(128496))
('Eddard-Stark', 'Jorah-Mormont', np.int64(123056))
('Eddard-Stark', 'Irri', np.int64(120245))
('Eddard-Stark', 'Mirri-Maz-Duur', np.int64(120217))
('Eddard-Stark', 'Jhiqui', np.int64(120072))
('Eddard-Stark', 'Samwell-Tarly', np.int64(120044))
('Eddard-Stark', 'Qotho', np.int64(119914))
('Eddard-Stark', 'Bronn', np.int64(119801))
('Eddard-Stark', 'Waymar-Royce', np.int64(119800))
('Eddard-Stark', 'Will-(prologue)', np.int64(119766))
('Eddard-Stark', 'Doreah', np.int64(119733))
('Eddard-Stark', 'Gared', np.int64(119627))

```

Nodes with high Hamming Distance in Book 2:

```

('Tyrion-Lannister', 'Bran-Stark', np.int64(47199))
('Tyrion-Lannister', 'Jon-Snow', np.int64(41945))
('Bran-Stark', 'Joffrey-Baratheon', np.int64(38695))
('Tyrion-Lannister', 'Jeor-Mormont', np.int64(36978))
('Tyrion-Lannister', 'Arya-Stark', np.int64(36427))
('Tyrion-Lannister', 'Hodor', np.int64(36217))
('Tyrion-Lannister', 'Luwine', np.int64(35621))
('Tyrion-Lannister', 'Hot-Pie', np.int64(35579))
('Tyrion-Lannister', 'Theon-Greyjoy', np.int64(35325))
('Tyrion-Lannister', 'Samwell-Tarly', np.int64(35106))
('Tyrion-Lannister', 'Rickon-Stark', np.int64(34670))
('Tyrion-Lannister', 'Jojen-Reed', np.int64(34611))
('Tyrion-Lannister', 'Qhorin-Halfhand', np.int64(34455))
('Tyrion-Lannister', 'Lommy-Greenhands', np.int64(33956))
('Tyrion-Lannister', 'Gendry', np.int64(33876))

```

Nodes with high Hamming Distance in Book 3:

```

('Tyrion-Lannister', 'Jon-Snow', np.int64(50752))
('Joffrey-Baratheon', 'Jon-Snow', np.int64(41524))
('Tyrion-Lannister', 'Samwell-Tarly', np.int64(38549))
('Jaime-Lannister', 'Jon-Snow', np.int64(38445))
('Tyrion-Lannister', 'Bran-Stark', np.int64(38209))

```

```
('Jon-Snow', 'Sansa-Stark', np.int64(37550))
('Tyrion-Lannister', 'Mance-Rayder', np.int64(36298))
('Tyrion-Lannister', 'Daenerys-Targaryen', np.int64(35546))
('Tyrion-Lannister', 'Hodor', np.int64(35095))
('Jon-Snow', 'Bran-Stark', np.int64(34287))
('Tyrion-Lannister', 'Meera-Reed', np.int64(33584))
('Tyrion-Lannister', 'Jojen-Reed', np.int64(32848))
('Jon-Snow', 'Arya-Stark', np.int64(32644))
('Tyrion-Lannister', 'Grenn', np.int64(32505))
('Tyrion-Lannister', 'Davos-Seaworth', np.int64(32401))
```

Nodes with high Hamming Distance in Book 4:

```
('Cersei-Lannister', 'Samwell-Tarly', np.int64(29685))
('Cersei-Lannister', 'Euron-Greyjoy', np.int64(27533))
('Cersei-Lannister', 'Victarion-Greyjoy', np.int64(27029))
('Cersei-Lannister', 'Gilly', np.int64(26579))
('Cersei-Lannister', 'Aemon-Targaryen-(Maester-Aemon)', np.int64(26499))
('Cersei-Lannister', 'Podrick-Payne', np.int64(26210))
('Cersei-Lannister', 'Arianne-Martell', np.int64(26033))
('Cersei-Lannister', 'Aeron-Greyjoy', np.int64(25827))
('Cersei-Lannister', 'Jon-Snow', np.int64(25759))
('Cersei-Lannister', 'Aero-Hotah', np.int64(25721))
('Cersei-Lannister', 'Doran-Martell', np.int64(25599))
('Cersei-Lannister', 'Petyr-Baelish', np.int64(25296))
('Cersei-Lannister', 'Brienne-of-Tarth', np.int64(25237))
('Cersei-Lannister', 'Balon-Greyjoy', np.int64(25222))
('Cersei-Lannister', 'Arya-Stark', np.int64(25096))
```

Nodes with high Hamming Distance in Book 5:

```
('Daenerys-Targaryen', 'Jon-Snow', np.int64(48124))
('Daenerys-Targaryen', 'Theon-Greyjoy', np.int64(37016))
('Daenerys-Targaryen', 'Stannis-Baratheon', np.int64(36679))
('Daenerys-Targaryen', 'Tyrion-Lannister', np.int64(36639))
('Daenerys-Targaryen', 'Ramsay-Snow', np.int64(35588))
('Jon-Snow', 'Hizdahr-zo-Loraq', np.int64(33839))
('Tyrion-Lannister', 'Jon-Snow', np.int64(32947))
('Daenerys-Targaryen', 'Bran-Stark', np.int64(31622))
('Daenerys-Targaryen', 'Melisandre', np.int64(30964))
('Daenerys-Targaryen', 'Hodor', np.int64(30335))
('Daenerys-Targaryen', 'Tormund', np.int64(30170))
('Jon-Snow', 'Barristan-Selmy', np.int64(30019))
('Daenerys-Targaryen', 'Bowen-Marsh', np.int64(29881))
('Daenerys-Targaryen', 'Selyse-Florent', np.int64(29609))
('Daenerys-Targaryen', 'Meera-Reed', np.int64(29282))
```

Nodes with high Hamming Distance in all Books:

```
('Daenerys-Targaryen', 'Jon-Snow', np.int64(55061))
('Daenerys-Targaryen', 'Tyrion-Lannister', np.int64(49692))
```

```

('Jon-Snow', 'Tyrion-Lannister', np.int64(49299))
('Daenerys-Targaryen', 'Arya-Stark', np.int64(41548))
('Jon-Snow', 'Hizzahr-zo-Loraq', np.int64(40859))
('Daenerys-Targaryen', 'Stannis-Baratheon', np.int64(40410))
('Daenerys-Targaryen', 'Theon-Greyjoy', np.int64(39699))
('Daenerys-Targaryen', 'Bran-Stark', np.int64(39507))
('Daenerys-Targaryen', 'Ramsay-Snow', np.int64(37657))
('Jon-Snow', 'Arya-Stark', np.int64(37431))
('Jon-Snow', 'Barristan-Selmy', np.int64(37273))
('Jaime-Lannister', 'Daenerys-Targaryen', np.int64(35701))
('Daenerys-Targaryen', 'Cersei-Lannister', np.int64(35531))
('Jon-Snow', 'Quentyn-Martell', np.int64(35467))
('Jon-Snow', 'Theon-Greyjoy', np.int64(35184))

[ ]: print("Nodes with low Hamming Distance in Book 1: ")
for el in bottom_hamming_1:
    print(el)

print("\nNodes with low Hamming Distance in Book 2: ")
for el in bottom_hamming_2:
    print(el)

print("\nNodes with low Hamming Distance in Book 3: ")
for el in bottom_hamming_3:
    print(el)

print("\nNodes with low Hamming Distance in Book 4: ")
for el in bottom_hamming_4:
    print(el)

print("\nNodes with low Hamming Distance in Book 5: ")
for el in bottom_hamming_5:
    print(el)

print("\nNodes with low Hamming Distance in all Books: ")
for el in bottom_hamming_all:
    print(el)

```

Nodes with low Hamming Distance in Book 1:

```

('Kurleket', 'Leo-Lefford', np.int64(1))
('Donnel-Waynwood', 'Moreo-Tumitis', np.int64(0))
('Donnel-Waynwood', 'Mychel-Redfort', np.int64(0))
('Moreo-Tumitis', 'Mychel-Redfort', np.int64(0))
('Chella', 'Kurleket', np.int64(0))
('Gendry', 'Howland-Reed', np.int64(0))
('Gendry', 'Wylla', np.int64(0))
('Howland-Reed', 'Wylla', np.int64(0))
('Jacks', 'Porther', np.int64(0))

```

```
('Jacks', 'Tregar', np.int64(0))
('Jacks', 'Varly', np.int64(0))
('Porther', 'Tregar', np.int64(0))
('Porther', 'Varly', np.int64(0))
('Tregar', 'Varly', np.int64(0))
('Hobb', 'Othor', np.int64(0))
```

Nodes with low Hamming Distance in Book 2:

```
('Howland-Reed', 'Poxy-Tym', np.int64(0))
('Captain-Myraham', 'Euron-Greyjoy', np.int64(0))
('Captain-Myraham', 'Jason-Mallister', np.int64(0))
('Captain-Myraham', 'Kyra', np.int64(0))
('Doreah', 'Lynesse-Hightower', np.int64(0))
('Doreah', 'Mirri-Maz-Duur', np.int64(0))
('Lynesse-Hightower', 'Mirri-Maz-Duur', np.int64(0))
('Imry-Florent', 'Matthos-Seaworth', np.int64(0))
('Euron-Greyjoy', 'Jason-Mallister', np.int64(0))
('Euron-Greyjoy', 'Kyra', np.int64(0))
('Hake', 'Lark', np.int64(0))
('High-Septon-(fat_one)', 'High-Septon-(Tyrions)', np.int64(0))
('Jason-Mallister', 'Kyra', np.int64(0))
('Martyn-Rivers', 'Rickard-Karstark', np.int64(0))
('Robert-Arryn', 'Trystane-Martell', np.int64(0))
```

Nodes with low Hamming Distance in Book 3:

```
('Kella', 'Tysha', np.int64(9))
('Merrett-Frey', 'Yohn-Royce', np.int64(9))
('Mallador-Locke', 'Ottyn-Wythers', np.int64(4))
('Perwyn-Frey', 'Rolph-Spicer', np.int64(4))
('Rolph-Spicer', 'Tristifer-IV-Mudd', np.int64(4))
('Chataya', 'Galyeon-of-Cuy', np.int64(1))
('Clubfoot-Karl', 'Dirk', np.int64(1))
('Grolio', 'Rhaegel-Targaryen', np.int64(1))
('Kella', 'Mordane', np.int64(1))
('Arson', 'Zei', np.int64(0))
('Mad-Huntsman', 'Pate-(Old)', np.int64(0))
('Brenett', 'Rodrik-Cassel', np.int64(0))
('Daeron-I-Targaryen', 'Patchface', np.int64(0))
('Grisel', 'Yohn-Royce', np.int64(0))
('Perwyn-Frey', 'Tristifer-IV-Mudd', np.int64(0))
```

Nodes with low Hamming Distance in Book 4:

```
('Garrett-Paege', 'Shitmouth', np.int64(0))
('Garrett-Paege', 'Wylis-Manderly', np.int64(0))
('Hugo-Vance', 'Joy-Hill', np.int64(0))
('Hugo-Vance', 'Lewys-Piper', np.int64(0))
('Hugo-Vance', 'Shitmouth', np.int64(0))
('Hugo-Vance', 'Wylis-Manderly', np.int64(0))
```

```
('Joy-Hill', 'Lewys-Piper', np.int64(0))
('Joy-Hill', 'Shitmouth', np.int64(0))
('Joy-Hill', 'Wylis-Manderly', np.int64(0))
('Karyl-Vance', 'Ronnet-Connington', np.int64(0))
('Karyl-Vance', 'Sybell-Spicer', np.int64(0))
('Lewys-Piper', 'Shitmouth', np.int64(0))
('Lewys-Piper', 'Wylis-Manderly', np.int64(0))
('Ronnet-Connington', 'Sybell-Spicer', np.int64(0))
('Shitmouth', 'Wylis-Manderly', np.int64(0))
```

Nodes with low Hamming Distance in Book 5:

```
('Daeron-II-Targaryen', 'Lysono-Maar', np.int64(0))
('Eroeh', 'Ezzara', np.int64(0))
('Eroeh', 'Lysono-Maar', np.int64(0))
('Ezzara', 'Lysono-Maar', np.int64(0))
('Soren-Shieldbreaker', 'Ulmer', np.int64(0))
('Longwater-Pyke', 'Ralf-(Limper)', np.int64(0))
('Longwater-Pyke', 'Wulfe', np.int64(0))
('Lyanna-Mormont', 'Marya-Seaworth', np.int64(0))
('Lyanna-Mormont', 'Roose-Ryswell', np.int64(0))
('Lyanna-Mormont', 'William-Foxglove', np.int64(0))
('Marya-Seaworth', 'Roose-Ryswell', np.int64(0))
('Marya-Seaworth', 'William-Foxglove', np.int64(0))
('Morgo', 'Shrouded-Lord', np.int64(0))
('Ralf-(Limper)', 'Wulfe', np.int64(0))
('Roose-Ryswell', 'William-Foxglove', np.int64(0))
```

Nodes with low Hamming Distance in all Books:

```
('Ralf-(Limper)', 'Longwater-Pyke', np.int64(0))
('Ralf-(Limper)', 'Wulfe', np.int64(0))
('Rolfe-the-Dwarf', 'Torgon-Greyiron', np.int64(0))
('Larraq', 'Widower', np.int64(0))
('Barsena', 'Yurkhaz-zo-Yunzak', np.int64(0))
('Corliss-Penny', 'Lyanna-Mormont', np.int64(0))
('Corliss-Penny', 'Roose-Ryswell', np.int64(0))
('Corliss-Penny', 'William-Foxglove', np.int64(0))
('Ezzara', 'Lysono-Maar', np.int64(0))
('Soren-Shieldbreaker', 'Ulmer', np.int64(0))
('Longwater-Pyke', 'Wulfe', np.int64(0))
('Lyanna-Mormont', 'Roose-Ryswell', np.int64(0))
('Lyanna-Mormont', 'William-Foxglove', np.int64(0))
('Morgo', 'Shrouded-Lord', np.int64(0))
('Roose-Ryswell', 'William-Foxglove', np.int64(0))
```

La distanza di Hamming, e quindi anche quella Euclidea sono maggiori tra i personaggi principali. Mentre è nulla tra le comparse. I motivi sono probabilmente gli stessi visti in precedenza.

Regular Equivalence Regular equivalence misura la similarità strutturale tra nodi: due nodi sono “equivalenti” se i loro vicini lo sono a loro volta.

L'algoritmo: 1. Calcola α per garantire convergenza. 2. Inizializza $\sigma = I$ (ogni nodo perfettamente simile a sé stesso). 3. Itera $\sigma \leftarrow \alpha \cdot A \cdot \sigma \cdot A^T$, ripristinando la diagonale a 1, fino a convergenza. 4. Restituisce la matrice σ e, su richiesta, le top-N coppie più simili.

```
[ ]: def calculate_regular_equivalence(graph, alpha=None, max_iter=100, tol=1e-6):
    """
    Calcola la matrice di 'regular equivalence' per il grafo non orientato
    `graph`.

    Restituisce:
        - sigma: numpy array n×n con i punteggi _ij
        - nodes: lista dei nodi, in corrispondenza con gli indici di sigma
    """

    # Matrice di adiacenza (dense)
    A = nx.to_numpy_array(graph, dtype=float)
    n = A.shape[0]

    # Se non passo alpha, lo scelgo in modo da garantire convergenza:
    # l'operatore  $X \rightarrow A X A$  ha raggio spettrale  $\max(A)^2$ ,
    # quindi imposto  $\alpha = 1 / \max(A)^2$ .
    vals = np.linalg.eigvals(A)
    rho = np.max(np.real(vals))
    alpha_opt = 1.0 / (rho * rho)

    if alpha is None:
        alpha = alpha_opt
    elif alpha > alpha_opt:
        print(f"Attenzione: il valore di alpha ({alpha:.3g}) è troppo grande, "
              f"potrebbe divergere. Lo riduco a {alpha_opt:.3g}.")
        alpha = alpha_opt

    # Inizializzo con l'identità (ognuno è perfettamente simile a sé stesso)
    sigma = np.eye(n, dtype=float)

    for _ in range(max_iter):
        sigma_prev = sigma

        # Iterazione vettoriale:  $\sigma \leftarrow \alpha \cdot A \cdot \sigma \cdot A^T$ 
        sigma = alpha * (A @ sigma_prev @ A.T)

        # Reimposto la diagonale a 1
        np.fill_diagonal(sigma, 1.0)

        # Verifico convergenza
        if np.linalg.norm(sigma - sigma_prev, ord='fro') < tol:
            break
```

```

    return sigma, list(graph.nodes())

[ ]: def get_similarity_results(sigma, nodes):
    """
    Converte la matrice sigma e la lista di nodi in una lista di triple:
    (nodo_i, nodo_j, _ij) per ogni i != j
    """
    results = []
    n = sigma.shape[0]
    for i in range(n):
        for j in range(n):
            if i != j:
                results.append((nodes[i], nodes[j], sigma[i, j]))
    return results

[ ]: def top_n_similar_pairs(similarity_results, n):
    """
    Dalla lista di triple (u,v,score) restituisce le prime n coppie più simili.
    """
    sorted_results = sorted(similarity_results, key=lambda x: x[2],  

                           reverse=True)
    return sorted_results[:n]

[ ]: def top_n_for_graph(graph, n, alpha=None, max_iter=100, tol=1e-6):
    """
    Funzione helper: dato un grafo, restituisce direttamente
    le top-n coppie secondo regular-equivalence.
    """
    sigma, nodes = calculate_regular_equivalence(graph, alpha, max_iter, tol)
    sims = get_similarity_results(sigma, nodes)
    return top_n_similar_pairs(sims, n)

[ ]: top15_G1 = top_n_for_graph(G_1, 15)

print("Top 15 coppie secondo regular-equivalence per libro 1:")
for u, v, score in top15_G1:
    print(f" {u} {v}: {score:.4f}")

```

Top 15 coppie secondo regular-equivalence per libro 1:

Eddard-Stark	Robert-Baratheon	1.9170
Robert-Baratheon	Eddard-Stark	1.9170
Eddard-Stark	Cersei-Lannister	0.8069
Cersei-Lannister	Eddard-Stark	0.8069
Robert-Baratheon	Cersei-Lannister	0.7098
Cersei-Lannister	Robert-Baratheon	0.7098
Sansa-Stark	Eddard-Stark	0.6848

```
Eddard-Stark Sansa-Stark: 0.6848
Catelyn-Stark Eddard-Stark: 0.6691
Eddard-Stark Catelyn-Stark: 0.6691
Petyr-Baelish Eddard-Stark: 0.6471
Eddard-Stark Petyr-Baelish: 0.6471
Jon-Snow Eddard-Stark: 0.6291
Eddard-Stark Jon-Snow: 0.6291
Eddard-Stark Joffrey-Baratheon: 0.6140
```

```
[ ]: top15_G2 = top_n_for_graph(G_2, 15)

print("Top 15 coppie secondo regular-equivalence per libro 2:")
for u, v, score in top15_G2:
    print(f" {u} {v}: {score:.4f}")
```

```
Top 15 coppie secondo regular-equivalence per libro 2:
Tyrion-Lannister Joffrey-Baratheon: 1.6561
Joffrey-Baratheon Tyrion-Lannister: 1.6561
Tyrion-Lannister Cersei-Lannister: 1.4512
Cersei-Lannister Tyrion-Lannister: 1.4512
Joffrey-Baratheon Cersei-Lannister: 1.2349
Cersei-Lannister Joffrey-Baratheon: 1.2349
Sansa-Stark Tyrion-Lannister: 0.9434
Tyrion-Lannister Sansa-Stark: 0.9434
Tyrion-Lannister Stannis-Baratheon: 0.9158
Stannis-Baratheon Tyrion-Lannister: 0.9158
Joffrey-Baratheon Sansa-Stark: 0.8549
Sansa-Stark Joffrey-Baratheon: 0.8549
Varys Tyrion-Lannister: 0.7916
Tyrion-Lannister Varys: 0.7916
Stannis-Baratheon Joffrey-Baratheon: 0.7849
```

```
[ ]: top15_G3 = top_n_for_graph(G_3, 15)

print("Top 15 coppie secondo regular-equivalence per libro 3:")
for u, v, score in top15_G3:
    print(f" {u} {v}: {score:.4f}")
```

```
Top 15 coppie secondo regular-equivalence per libro 3:
Tyrion-Lannister Joffrey-Baratheon: 1.7327
Joffrey-Baratheon Tyrion-Lannister: 1.7327
Tyrion-Lannister Sansa-Stark: 1.4203
Sansa-Stark Tyrion-Lannister: 1.4203
Sansa-Stark Joffrey-Baratheon: 1.1646
Joffrey-Baratheon Sansa-Stark: 1.1646
Tyrion-Lannister Cersei-Lannister: 1.1391
Cersei-Lannister Tyrion-Lannister: 1.1391
Tyrion-Lannister Jaime-Lannister: 1.0683
```

```
Jaime-Lannister  Tyrion-Lannister: 1.0683
Cersei-Lannister  Joffrey-Baratheon: 0.9397
Joffrey-Baratheon  Cersei-Lannister: 0.9397
Joffrey-Baratheon  Jaime-Lannister: 0.8912
Jaime-Lannister  Joffrey-Baratheon: 0.8912
Tyrion-Lannister  Tywin-Lannister: 0.7943
```

```
[ ]: top15_G4 = top_n_for_graph(G_4, 15)

print("Top 15 coppie secondo regular-equivalence per libro 4:")
for u, v, score in top15_G4:
    print(f"  {u}  {v}: {score:.4f}")
```

```
Top 15 coppie secondo regular-equivalence per libro 4:
Tommen-Baratheon  Cersei-Lannister: 1.3829
Cersei-Lannister  Tommen-Baratheon: 1.3829
Margaery-Tyrell  Cersei-Lannister: 1.2198
Cersei-Lannister  Margaery-Tyrell: 1.2198
Cersei-Lannister  Jaime-Lannister: 1.0836
Jaime-Lannister  Cersei-Lannister: 1.0836
Margaery-Tyrell  Tommen-Baratheon: 0.7168
Tommen-Baratheon  Margaery-Tyrell: 0.7168
Tommen-Baratheon  Jaime-Lannister: 0.6486
Jaime-Lannister  Tommen-Baratheon: 0.6486
Taena-of-Myr  Cersei-Lannister: 0.6391
Cersei-Lannister  Taena-of-Myr: 0.6391
Margaery-Tyrell  Jaime-Lannister: 0.5787
Jaime-Lannister  Margaery-Tyrell: 0.5787
Tyrion-Lannister  Cersei-Lannister: 0.5700
```

```
[ ]: top15_G5 = top_n_for_graph(G_5, 15)

print("Top 15 coppie secondo regular-equivalence per libro 5:")
for u, v, score in top15_G5:
    print(f"  {u}  {v}: {score:.4f}")
```

```
Top 15 coppie secondo regular-equivalence per libro 5:
Daenerys-Targaryen  Hizdahr-zo-Loraq: 1.3754
Hizdahr-zo-Loraq  Daenerys-Targaryen: 1.3754
Barristan-Selmy  Daenerys-Targaryen: 1.2135
Daenerys-Targaryen  Barristan-Selmy: 1.2135
Quentyn-Martell  Daenerys-Targaryen: 0.8668
Daenerys-Targaryen  Quentyn-Martell: 0.8668
Barristan-Selmy  Hizdahr-zo-Loraq: 0.7419
Hizdahr-zo-Loraq  Barristan-Selmy: 0.7419
Daario-Naharis  Daenerys-Targaryen: 0.7218
Daenerys-Targaryen  Daario-Naharis: 0.7218
Skahaz-mo-Kandaq  Daenerys-Targaryen: 0.6628
```

```

Daenerys-Targaryen Skahaz-mo-Kandaq: 0.6628
Quentyn-Martell Hizdahr-zo-Loraq: 0.5195
Hizdahr-zo-Loraq Quentyn-Martell: 0.5195
Daenerys-Targaryen Missandei: 0.4753

```

6.1.8 Homophily-Assortative Mixing

L'assortative coefficient è una misura che indica la tendenza dei nodi in una rete a connettersi con altri nodi simili o dissimili. Un valore positivo indica che i nodi tendono a connettersi con nodi simili (assortative mixing), mentre un valore negativo indica che i nodi tendono a connettersi con nodi dissimili (disassortative mixing).

```

[ ]: r_1 = nx.degree_assortativity_coefficient(G_1, weight='weight')
r_2 = nx.degree_assortativity_coefficient(G_2, weight='weight')
r_3 = nx.degree_assortativity_coefficient(G_3, weight='weight')
r_4 = nx.degree_assortativity_coefficient(G_4, weight='weight')
r_5 = nx.degree_assortativity_coefficient(G_5, weight='weight')
r_all = nx.degree_assortativity_coefficient(G_all, weight='weight')

[ ]: print(f"The assortative coefficient for Book 1 is: {r_1:.3f}")
print(f"The assortative coefficient for Book 2 is: {r_2:.3f}")
print(f"The assortative coefficient for Book 3 is: {r_3:.3f}")
print(f"The assortative coefficient for Book 4 is: {r_4:.3f}")
print(f"The assortative coefficient for Book 5 is: {r_5:.3f}")
print(f"The assortative coefficient for all Books is: {r_all:.3f}")

```

```

The assortative coefficient for Book 1 is: -0.166
The assortative coefficient for Book 2 is: -0.125
The assortative coefficient for Book 3 is: -0.133
The assortative coefficient for Book 4 is: -0.137
The assortative coefficient for Book 5 is: -0.185
The assortative coefficient for all Books is: -0.133

```

Tutti i coefficienti sono negativi. Questo vuol dire che i nodi tendono a collegarsi con nodi che hanno grado diverso rispetto al loro.

Se ci pensiamo ha senso: spesso i nodi con grado alto sono legati a nodi con grado basso creando dei cluster che hanno come centro dei personaggi principali. Quindi si tendono a costruirsi delle strutture a stella nel grafo.

Il dato ha anche molto senso se rapportato alla trama che spazia in zone geografiche diverse di un mondo molto ampio con tantissimi personaggi, di cui molti sono comparse o personaggi secondari che ruotano intorno alle vicende di pochi personaggi principali.

Il fatto che il libro 5 sia quello con assortatività più bassa ha senso: qui Tyrion uccide suo padre e scappa da Approdo del Re. Quindi ci sono diverse storie tutte ambientate in punti diversi: la storia di Jon Snow alla barriera, Tyrion che fugge, Daenerys che ottiene sempre più potere e le vicende intorno a Stannis Baratheon al nord.

Mentre il libro 2 è quello più assortativo perché quasi tutta la trama tende a concentrarsi in un luogo solo: Approdo del Re. E molti dei personaggi nella storia hanno grado simile, con un numero

minore di comparse. Anche il grafo appare con meno formazioni a stella.

6.2 Network

6.2.1 Small World Effect

La small worldness può essere calcolata usando due valori: sigma e omega.

Di solito, una network è small world se $\sigma > 1$ oppure se ha un valore ω vicino allo 0.

```
[ ]: sigma_1 = nx.sigma(G_1, niter=1)
sigma_2 = nx.sigma(G_2, niter=1)
sigma_3 = nx.sigma(G_3, niter=1)
sigma_4 = nx.sigma(G_4, niter=1)
sigma_5 = nx.sigma(G_5, niter=1)
sigma_all = nx.sigma(G_all, niter=1)

omega_1 = nx.omega(G_1, niter=1)
omega_2 = nx.omega(G_2, niter=1)
omega_3 = nx.omega(G_3, niter=1)
omega_4 = nx.omega(G_4, niter=1)
omega_5 = nx.omega(G_5, niter=1)
omega_all = nx.omega(G_all, niter=1)
```

```
[ ]: if sigma_1 > 1:
    print(f"The graph for Book 1 is small world with sigma: {sigma_1:.3f} and
          omega: {omega_1:.3f}")
else:
    print(f"The graph for Book 1 is not small world with sigma: {sigma_1:.3f} and
          omega: {omega_1:.3f}")

if sigma_2 > 1:
    print(f"The graph for Book 2 is small world with sigma: {sigma_2:.3f} and
          omega: {omega_2:.3f}")
else:
    print(f"The graph for Book 2 is not small world with sigma: {sigma_2:.3f} and
          omega: {omega_2:.3f}")

if sigma_3 > 1:
    print(f"The graph for Book 3 is small world with sigma: {sigma_3:.3f} and
          omega: {omega_3:.3f}")
else:
    print(f"The graph for Book 3 is not small world with sigma: {sigma_3:.3f} and
          omega: {omega_3:.3f}")

if sigma_4 > 1:
    print(f"The graph for Book 4 is small world with sigma: {sigma_4:.3f} and
          omega: {omega_4:.3f}")
```

```

else:
    print(f"The graph for Book 4 is not small world with sigma: {sigma_4:.3f} and_
        omega: {omega_4:.3f}")

if sigma_5 > 1:
    print(f"The graph for Book 5 is small world with sigma: {sigma_5:.3f} and_
        omega: {omega_5:.3f}")
else:
    print(f"The graph for Book 5 is not small world with sigma: {sigma_5:.3f} and_
        omega: {omega_5:.3f}")

if sigma_all > 1:
    print(f"The graph for All Books is small world with sigma: {sigma_all:.3f}_
        and omega: {omega_all:.3f}")
else:
    print(f"The graph for All Books is not small world with sigma: {sigma_all:.
        3f} and omega: {omega_all:.3f}")

```

The graph for Book 1 is small world with sigma: 1.644 and omega: -0.073
The graph for Book 2 is small world with sigma: 2.209 and omega: -0.115
The graph for Book 3 is small world with sigma: 2.348 and omega: -0.110
The graph for Book 4 is small world with sigma: 2.403 and omega: -0.185
The graph for Book 5 is small world with sigma: 2.808 and omega: -0.115
The graph for All Books is small world with sigma: 2.314 and omega: -0.071

6.2.2 Degree Centrality

```
[ ]: def plot_degree_centrality_histogram_and_cdf(G, title=""):

    """
    Mostra l'istogramma della distribuzione della centralità di grado e la CDF
    in scala log-log.

    :param G: Grafo di NetworkX.
    :param title: Titolo del grafico.
    """

    # Calcolare la centralità di grado
    degree_centrality = nx.degree_centrality(G)

    # Estrarre i valori e calcolare le frequenze
    degree_values = list(degree_centrality.values())
    degree_values.sort()

    # Calcolare i gradi e le loro frequenze
    degree_counts = np.bincount([int(degree * (len(G) - 1)) for degree in
        degree_values])
    degrees = np.nonzero(degree_counts)[0]
    counts = degree_counts[degrees]
```

```

# Calcolare le frequenze cumulative
cumulative_frequencies = np.cumsum(counts) / np.sum(counts)

# Calcolare logaritmo dei gradi e delle frequenze
log_degrees = np.log(degrees[degrees > 0])
log_counts = np.log(counts[counts > 0])

# Eseguire la regressione lineare
slope, intercept, r_value, p_value, std_err = stats.linregress(log_degrees, log_counts)

# Calcolare alpha
alpha = -slope

# Creare la figura
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(16, 6))
fig.suptitle(title, fontsize=16, color="darkblue")

# Iстограмма
ax1.bar(degrees, counts, width=0.8, color='b', alpha=0.7)
ax1.set_xscale('log')
ax1.set_yscale('log')
ax1.set_title(f'Degree Centrality Distribution (Log-Log)', color="olive")
ax1.set_xlabel('Degree')
ax1.set_ylabel('Frequency')

# CDF
ax2.plot(degrees, cumulative_frequencies, linestyle='-', color='r')
ax2.set_title(f'Cumulative Distribution Function (Log-Log)', color="olive")
ax2.set_xlabel('Degree')
ax2.set_xscale('log')
ax2.set_yscale('log')
ax2.set_ylabel('Frequency')
ax2.grid(True, which="both", ls="--", linewidth=0.5)

plt.tight_layout()
plt.show()

return alpha

```

```

[ ]: alpha_1 = plot_degree_centrality_histogram_and_cdf(G_1, title="Degree Centrality Distribution for Book 1")

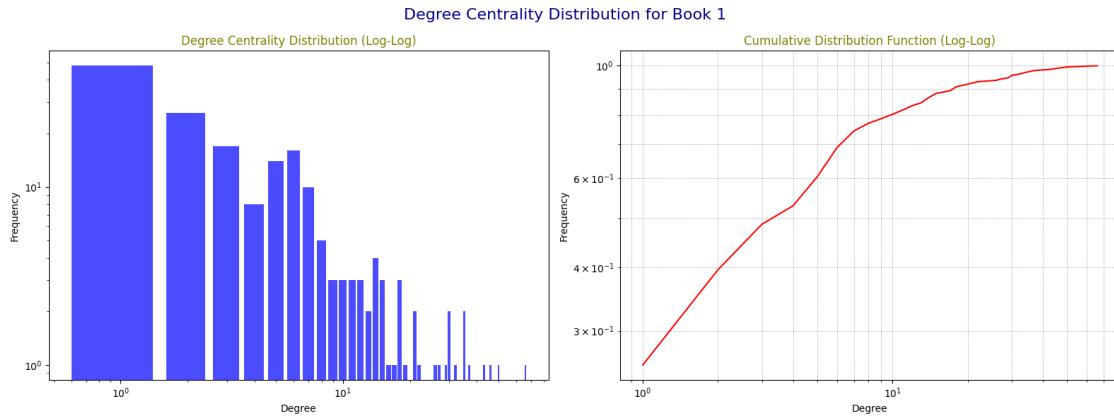
if alpha_1 >= 2 and alpha_1 <= 3:

```

```

    print(f"\n#### The value of alpha is: {alpha_1:.2}, so this network is a\u
→scale-free network ####")
else:
    print(f"\n#### The value of alpha is: {alpha_1:.2}, so this network is not\u
→a scale-free network ####")

```



```

#### The value of alpha is: 1.1, so this network is not a scale-free network
#####

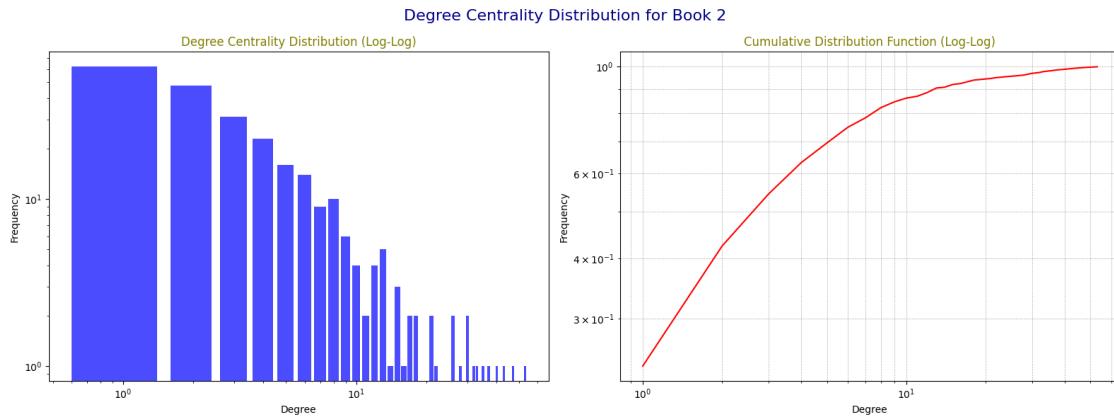
```

```

[ ]: alpha_2 = plot_degree_centrality_histogram_and_cdf(G_2, title="Degree\u
→Centrality Distribution for Book 2")

if alpha_2 >= 2 and alpha_2 <= 3:
    print(f"\n#### The value of alpha is: {alpha_2:.2}, so this network is a\u
→scale-free network ####")
else:
    print(f"\n#### The value of alpha is: {alpha_2:.2}, so this network is not\u
→a scale-free network ####")

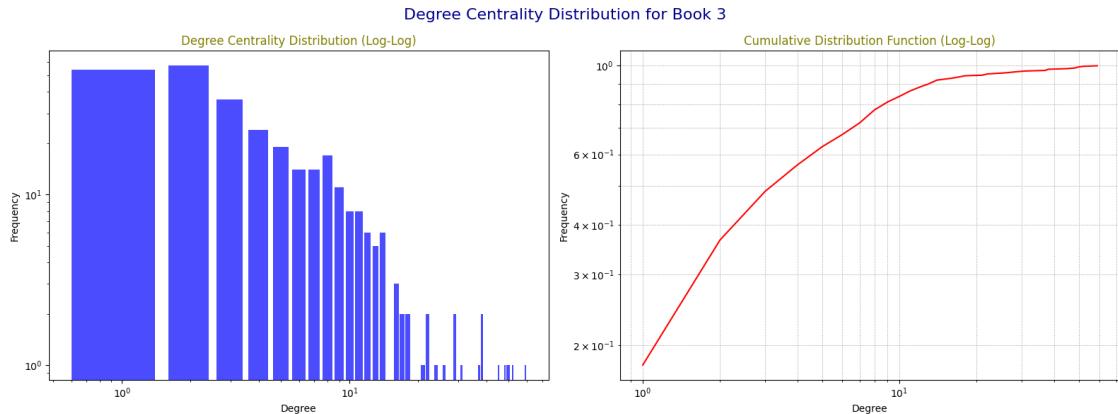
```



```
#### The value of alpha is: 1.3, so this network is not a scale-free network
####

[ ]: alpha_3 = plot_degree_centrality_histogram_and_cdf(G_3, title="Degree Centrality Distribution for Book 3")

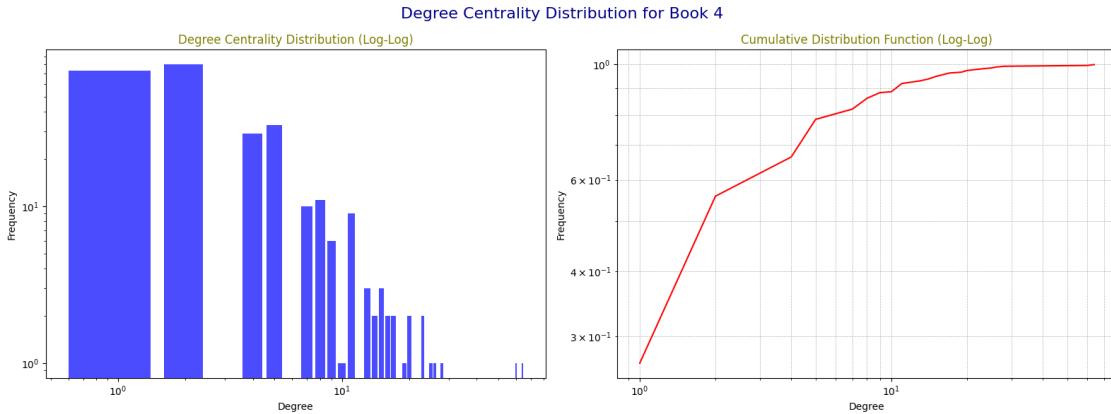
if alpha_3 >= 2 and alpha_3 <= 3:
    print(f"\n#### The value of alpha is: {alpha_3:.2}, so this network is a scale-free network ####")
else:
    print(f"\n#### The value of alpha is: {alpha_3:.2}, so this network is not a scale-free network ####")
```



```
#### The value of alpha is: 1.3, so this network is not a scale-free network
####
```

```
[ ]: alpha_4 = plot_degree_centrality_histogram_and_cdf(G_4, title="Degree Centrality Distribution for Book 4")

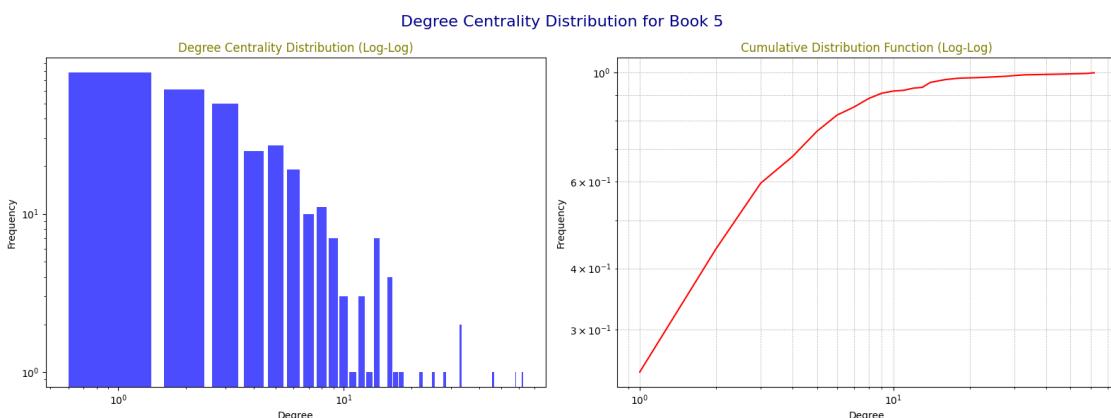
if alpha_4 >= 2 and alpha_4 <= 3:
    print(f"\n#### The value of alpha is: {alpha_4:.2}, so this network is a scale-free network ####")
else:
    print(f"\n#### The value of alpha is: {alpha_4:.2}, so this network is not a scale-free network ####")
```



```
#### The value of alpha is: 1.3, so this network is not a scale-free network
####
```

```
[ ]: alpha_5 = plot_degree_centrality_histogram_and_cdf(G_5, title="Degree_Centrality Distribution for Book 5")

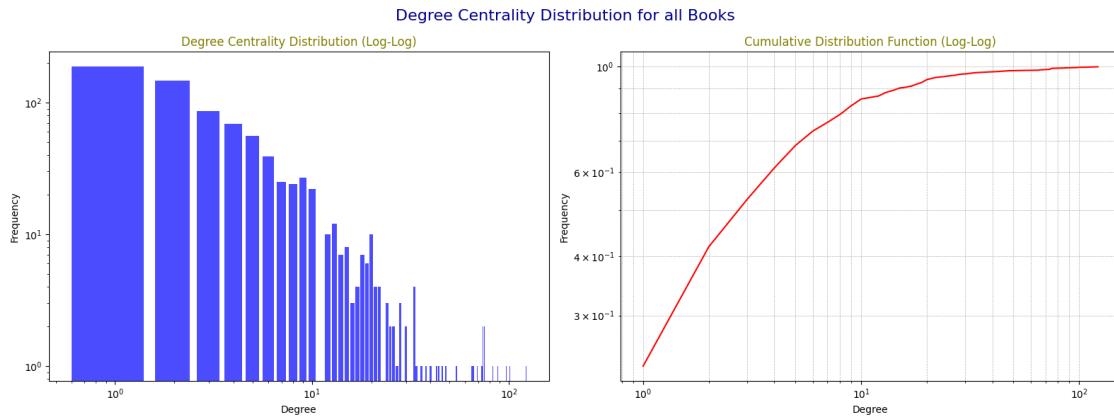
if alpha_5 >= 2 and alpha_5 <= 3:
    print(f"\n#### The value of alpha is: {alpha_5:.2}, so this network is a scale-free network ####")
else:
    print(f"\n#### The value of alpha is: {alpha_5:.2}, so this network is not a scale-free network ####")
```



```
#### The value of alpha is: 1.3, so this network is not a scale-free network
####
```

```
[ ]: alpha_all = plot_degree_centrality_histogram_and_cdf(G_all, title="Degree Centrality Distribution for all Books")

if alpha_all >= 2 and alpha_all <= 3:
    print(f"\n#### The value of alpha is: {alpha_all:.2}, so this network is a scale-free network ####")
else:
    print(f"\n#### The value of alpha is: {alpha_all:.2}, so this network is not a scale-free network ####")
```



```
#### The value of alpha is: 1.3, so this network is not a scale-free network ####
```

Quindi, dato che alpha non è compreso tra 2 e 3 allora la network non segue una power law.

Questo significa che NON si tratta di una Scale-free Network.

Ha molto senso: se togliamo a caso dei nodi dalla network rischiamo di perdere informazioni davvero importanti e la network rischia di collassare. Infatti la probabilità di eliminare a caso un nodo che rappresenta un personaggio importante della storia è molto elevata.

6.2.3 Eigenvector Centrality

```
[ ]: def plot_eigenvector_centrality_histogram_and_cdf(G, title=""):

    """
    Mostra l'istogramma della distribuzione della centralità degli autovettori e la CDF in scala log-log.

    :param G: Grafo di NetworkX.
    :param title: Titolo del grafico.
    """

    # Calcolare la centralità degli autovettori
    centrality = nx.eigenvector_centrality(G, weight='weight')
```

```

# Estrarre i valori e calcolare le frequenze
centrality_values = list(centrality.values())
centrality_values.sort()

# Calcolare le frequenze
centrality_counts = np.bincount([int(c * (len(G) - 1)) for c in
centrality_values])
centrality_values_nonzero = np.nonzero(centrality_counts)[0]

# Filtrare i valori zero
counts = centrality_counts[centrality_values_nonzero]
centrality_values_nonzero = centrality_values_nonzero[counts > 0]
counts = counts[counts > 0]

# Calcolare le frequenze cumulative
cumulative_frequencies = np.cumsum(counts) / np.sum(counts)

# Creare la figura
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(16, 6))
fig.suptitle(title, fontsize=16, color="darkblue")

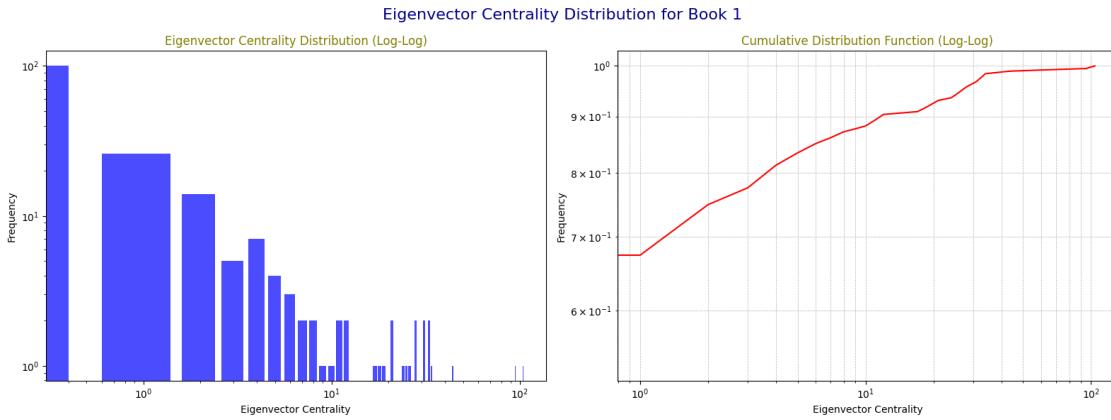
# Istogramma
ax1.bar(centrality_values_nonzero, counts, width=0.8, color='b', alpha=0.7)
ax1.set_xscale('log')
ax1.set_yscale('log')
ax1.set_title('Eigenvector Centrality Distribution (Log-Log)', color="olive")
ax1.set_xlabel('Eigenvector Centrality')
ax1.set_ylabel('Frequency')

# CDF
ax2.plot(centrality_values_nonzero, cumulative_frequencies, linestyle='--', color='r')
ax2.set_title('Cumulative Distribution Function (Log-Log)', color="olive")
ax2.set_xlabel('Eigenvector Centrality')
ax2.set_xscale('log')
ax2.set_yscale('log')
ax2.set_ylabel('Frequency')
ax2.grid(True, which="both", ls="--", linewidth=0.5)

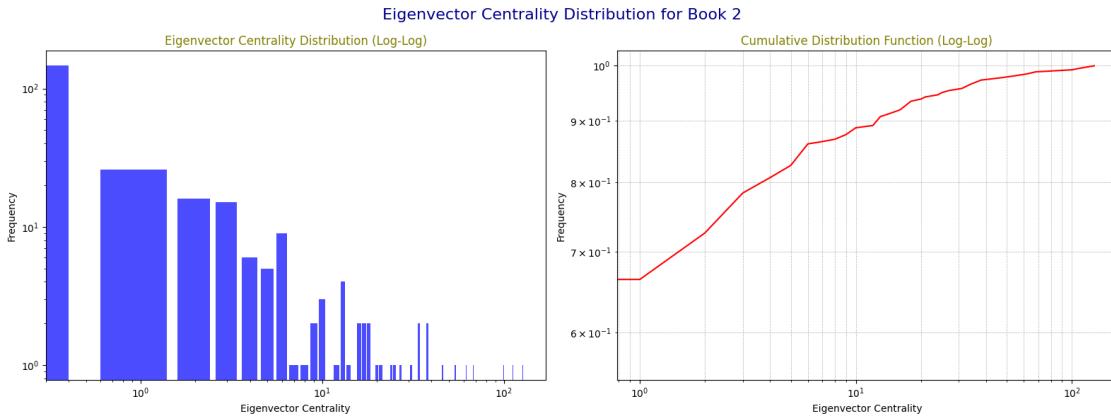
plt.tight_layout()
plt.show()

```

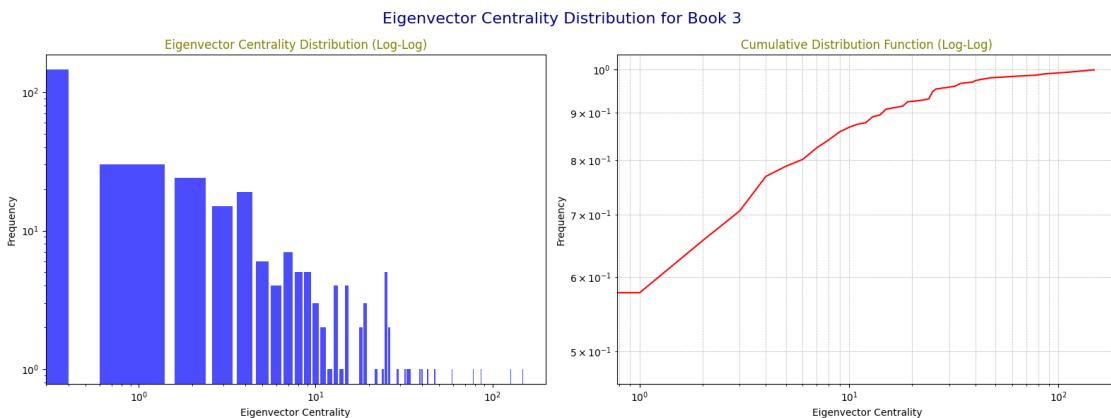
[]: plot_eigenvector_centrality_histogram_and_cdf(G_1, title="Eigenvector Centrality Distribution for Book 1")



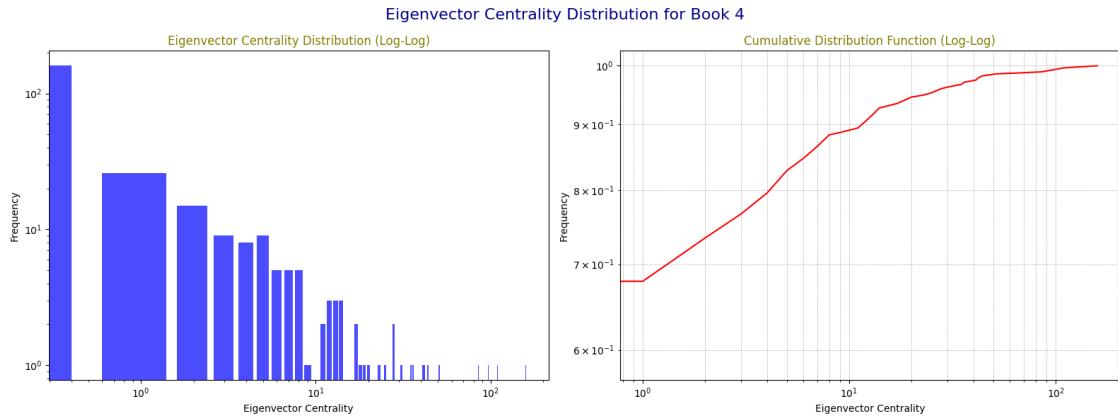
```
[ ]: plot_eigenvector_centrality_histogram_and_cdf(G_2, title="Eigenvector Centrality Distribution for Book 2")
```



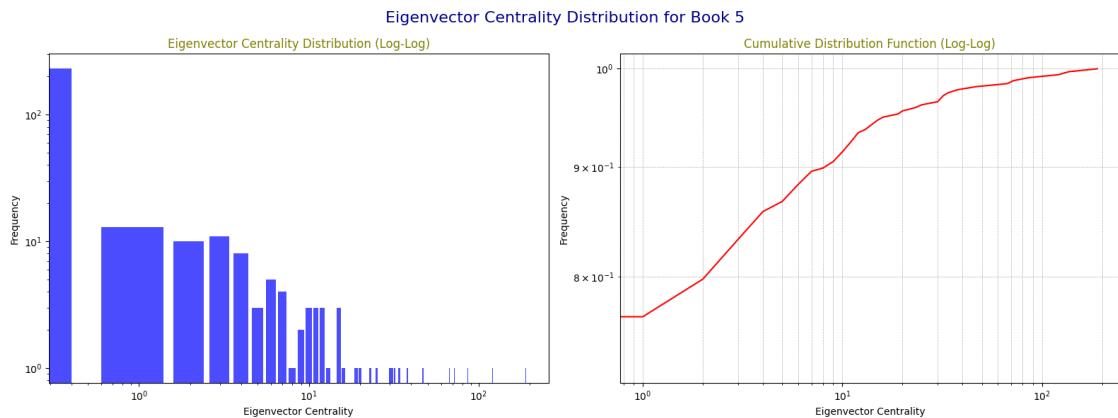
```
[ ]: plot_eigenvector_centrality_histogram_and_cdf(G_3, title="Eigenvector Centrality Distribution for Book 3")
```



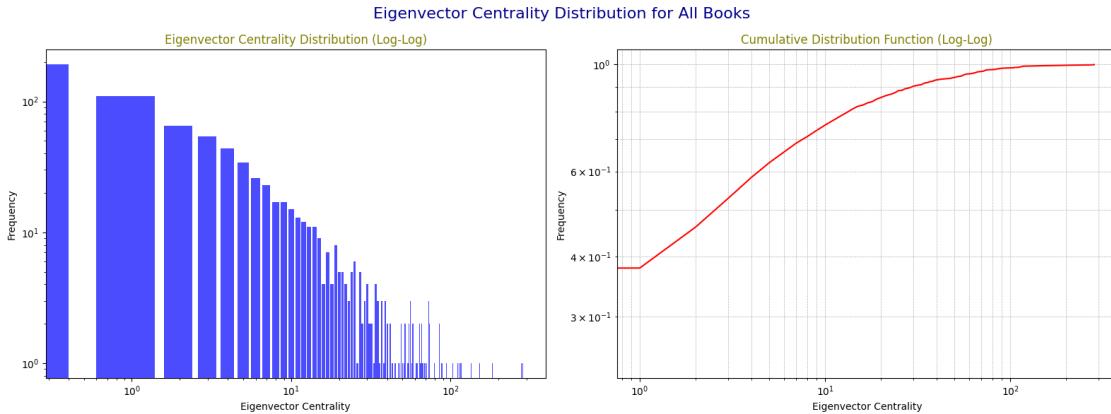
```
[ ]: plot_eigenvector_centrality_histogram_and_cdf(G_4, title="Eigenvector Centrality Distribution for Book 4")
```



```
[ ]: plot_eigenvector_centrality_histogram_and_cdf(G_5, title="Eigenvector Centrality Distribution for Book 5")
```



```
[ ]: plot_eigenvector_centrality_histogram_and_cdf(G_all, title="Eigenvector Centrality Distribution for All Books")
```



6.2.4 Closeness Centrality

```
[ ]: def plot_closeness_centrality_histogram_and_cdf(G, title=""):  
    """  
        Mostra l'istogramma della distribuzione della centralità degli autovettori  
        e la CDF in scala log-log.  
  
    :param G: Grafo di NetworkX.  
    :param title: Titolo del grafico.  
    """  
  
    # Calcolare la centralità degli autovettori  
    centrality = nx.closeness_centrality(G, u=None, distance=1)  
  
    # Estrarre i valori e calcolare le frequenze  
    centrality_values = list(centrality.values())  
    centrality_values.sort()  
  
    # Calcolare le frequenze  
    centrality_counts = np.bincount([int(c * (len(G) - 1)) for c in  
    centrality_values])  
    centrality_values_nonzero = np.nonzero(centrality_counts)[0]  
  
    # Filtrare i valori zero  
    counts = centrality_counts[centrality_values_nonzero]  
    centrality_values_nonzero = centrality_values_nonzero[counts > 0]  
    counts = counts[counts > 0]  
  
    # Calcolare le frequenze cumulative  
    cumulative_frequencies = np.cumsum(counts) / np.sum(counts)  
  
    # Creare la figura
```

```

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(16, 6))
fig.suptitle(title, fontsize=16, color="darkblue")

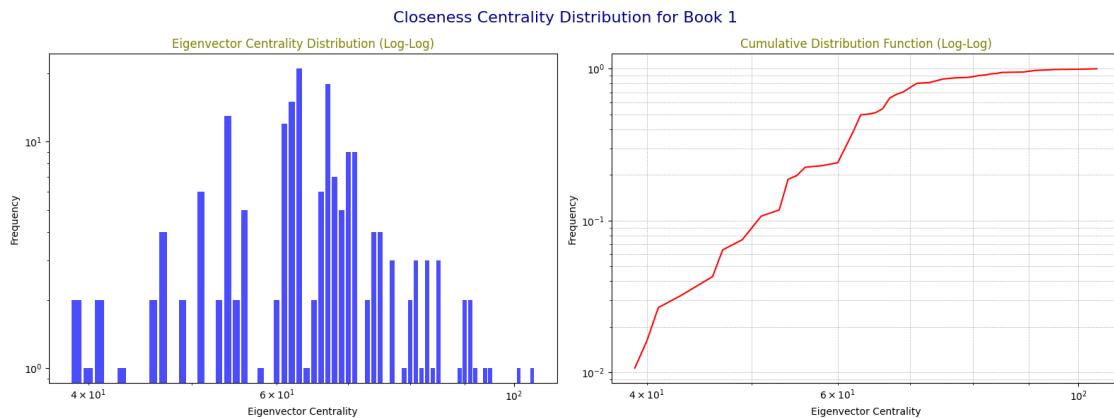
# Histogramma
ax1.bar(centrality_values_nonzero, counts, width=0.8, color='b', alpha=0.7)
ax1.set_xscale('log')
ax1.set_yscale('log')
ax1.set_title('Eigenvector Centrality Distribution (Log-Log)', color="olive")
ax1.set_xlabel('Eigenvector Centrality')
ax1.set_ylabel('Frequency')

# CDF
ax2.plot(centrality_values_nonzero, cumulative_frequencies, linestyle='-', color='r')
ax2.set_title('Cumulative Distribution Function (Log-Log)', color="olive")
ax2.set_xlabel('Eigenvector Centrality')
ax2.set_xscale('log')
ax2.set_yscale('log')
ax2.set_ylabel('Frequency')
ax2.grid(True, which="both", ls="--", linewidth=0.5)

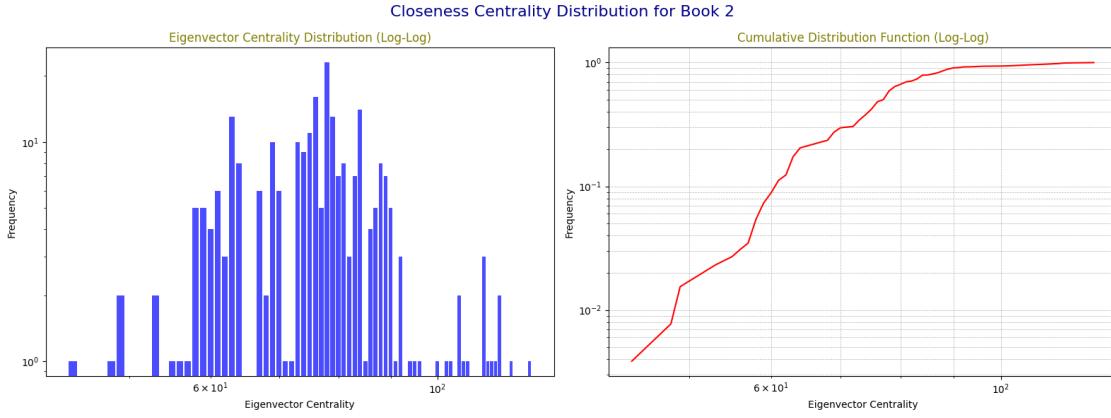
plt.tight_layout()
plt.show()

```

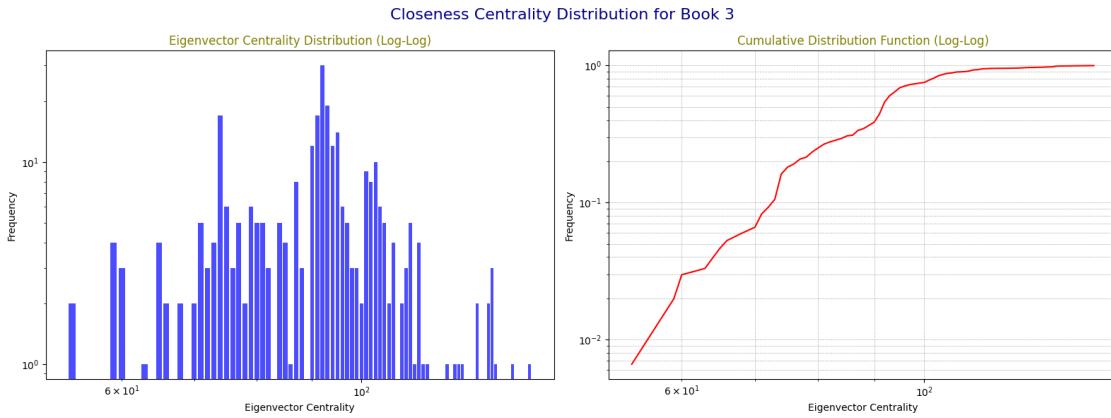
[]: plot_closeness_centrality_histogram_and_cdf(G_1, title="Closeness Centrality Distribution for Book 1")



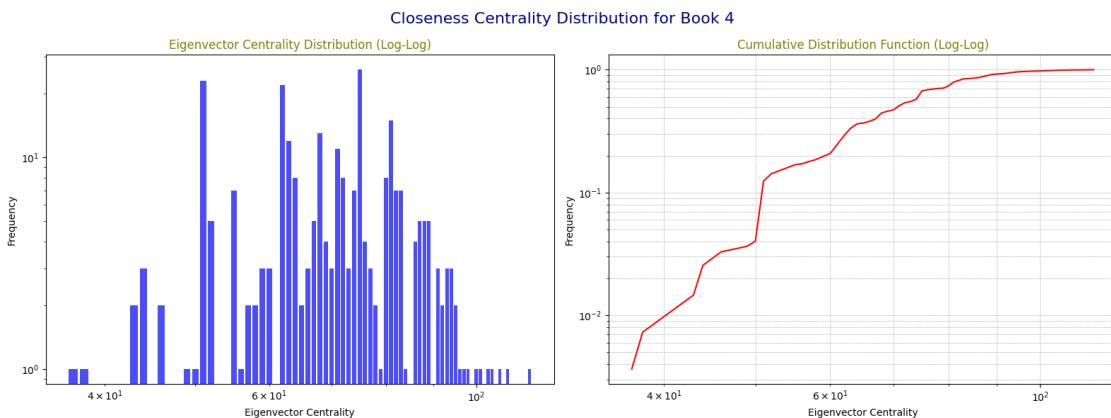
[]: plot_closeness_centrality_histogram_and_cdf(G_2, title="Closeness Centrality Distribution for Book 2")



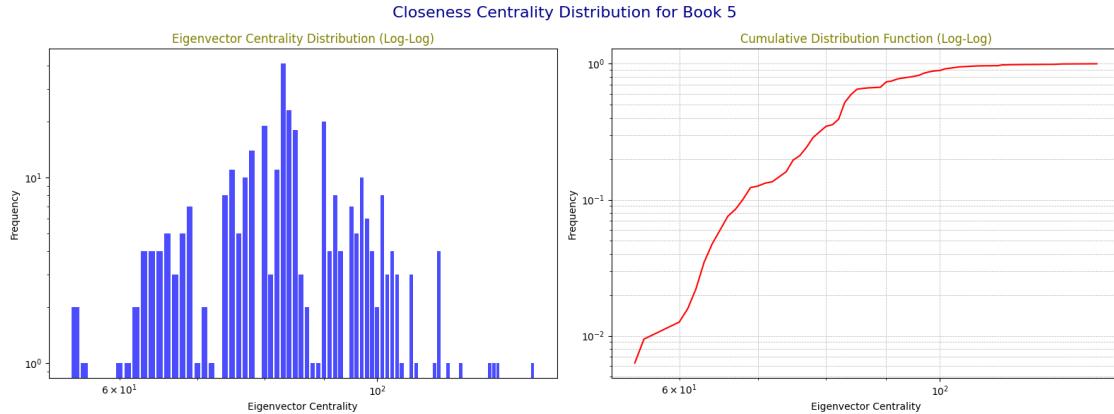
```
[ ]: plot_closeness_centrality_histogram_and_cdf(G_3, title="Closeness Centrality Distribution for Book 3")
```



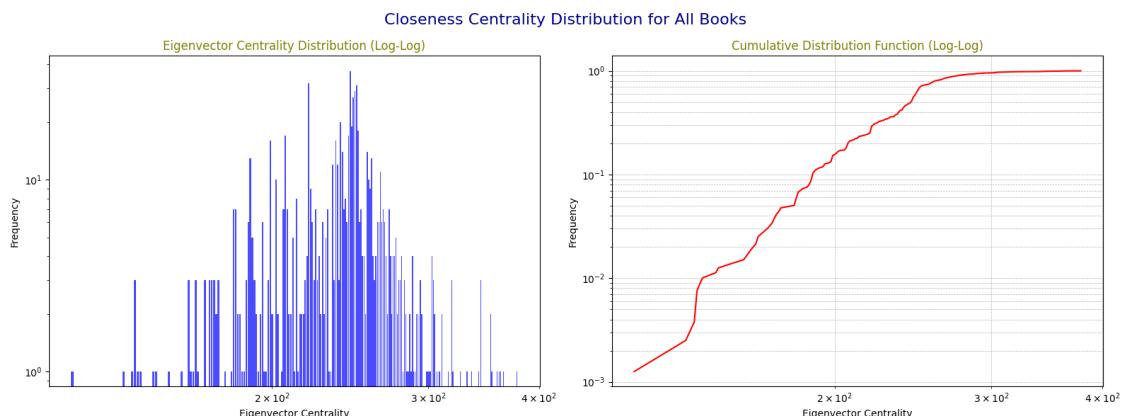
```
[ ]: plot_closeness_centrality_histogram_and_cdf(G_4, title="Closeness Centrality Distribution for Book 4")
```



```
[ ]: plot_closeness_centrality_histogram_and_cdf(G_5, title="Closeness Centrality Distribution for Book 5")
```



```
[ ]: plot_closeness_centrality_histogram_and_cdf(G_all, title="Closeness Centrality Distribution for All Books")
```



6.2.5 Betweenness Centrality

```
[ ]: def plot_betweenness_centrality_histogram_and_cdf(G, title=""):
    """
    Mostra l'istogramma della distribuzione della centralità degli autovettori e la CDF in scala log-log.

    :param G: Grafo di NetworkX.
    :param title: Titolo del grafico.
    """
```

```

"""
# Calcolare la centralità degli autovettori
centrality = nx.betweenness_centrality(G, weight='weight')

# Estrarre i valori e calcolare le frequenze
centrality_values = list(centrality.values())
centrality_values.sort()

# Calcolare le frequenze
centrality_counts = np.bincount([int(c * (len(G) - 1)) for c in
centrality_values])
centrality_values_nonzero = np.nonzero(centrality_counts)[0]

# Filtrare i valori zero
counts = centrality_counts[centrality_values_nonzero]
centrality_values_nonzero = centrality_values_nonzero[counts > 0]
counts = counts[counts > 0]

# Calcolare le frequenze cumulative
cumulative_frequencies = np.cumsum(counts) / np.sum(counts)

# Creare la figura
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(16, 6))
fig.suptitle(title, fontsize=16, color="darkblue")

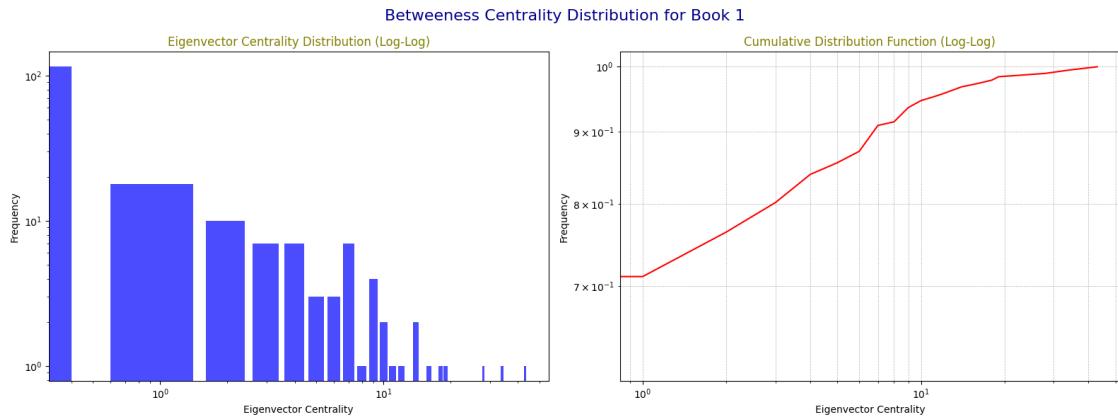
# Istogramma
ax1.bar(centrality_values_nonzero, counts, width=0.8, color='b', alpha=0.7)
ax1.set_xscale('log')
ax1.set_yscale('log')
ax1.set_title('Eigenvector Centrality Distribution (Log-Log)', color="olive")
ax1.set_xlabel('Eigenvector Centrality')
ax1.set_ylabel('Frequency')

# CDF
ax2.plot(centrality_values_nonzero, cumulative_frequencies, linestyle='--', color='r')
ax2.set_title('Cumulative Distribution Function (Log-Log)', color="olive")
ax2.set_xlabel('Eigenvector Centrality')
ax2.set_xscale('log')
ax2.set_yscale('log')
ax2.set_ylabel('Frequency')
ax2.grid(True, which="both", ls="--", linewidth=0.5)

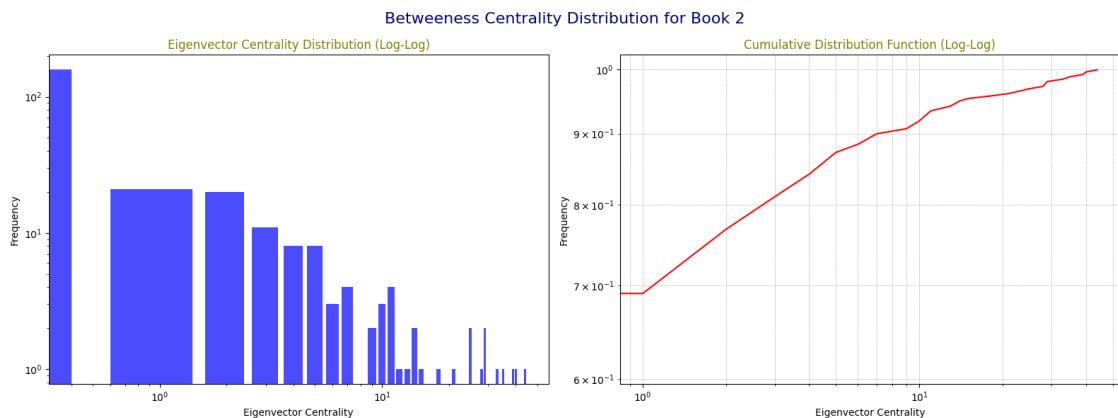
plt.tight_layout()
plt.show()

```

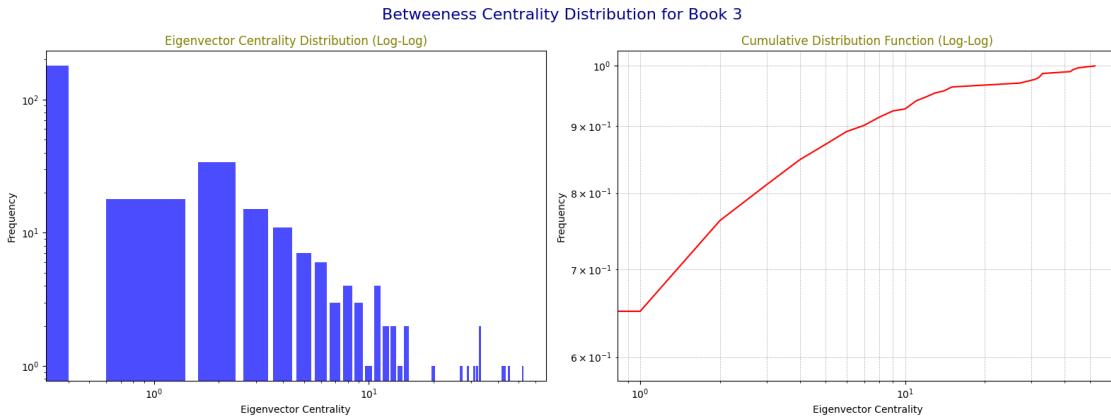
```
[ ]: plot_betweenness_centrality_histogram_and_cdf(G_1, title="Betweenness Centrality Distribution for Book 1")
```



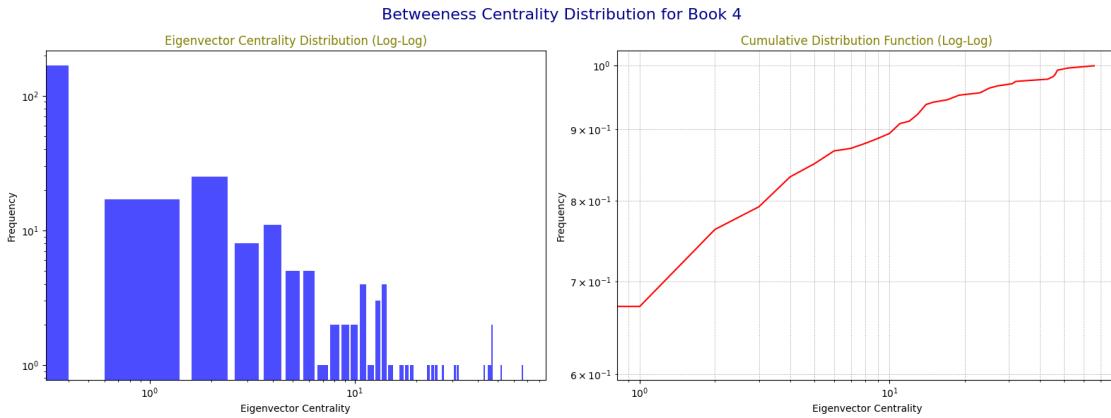
```
[ ]: plot_betweenness_centrality_histogram_and_cdf(G_2, title="Betweenness Centrality Distribution for Book 2")
```



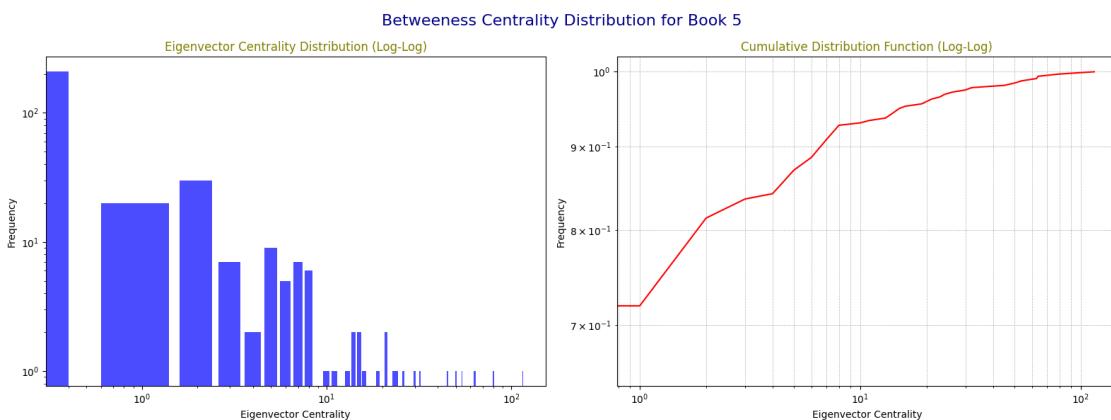
```
[ ]: plot_betweenness_centrality_histogram_and_cdf(G_3, title="Betweenness Centrality Distribution for Book 3")
```



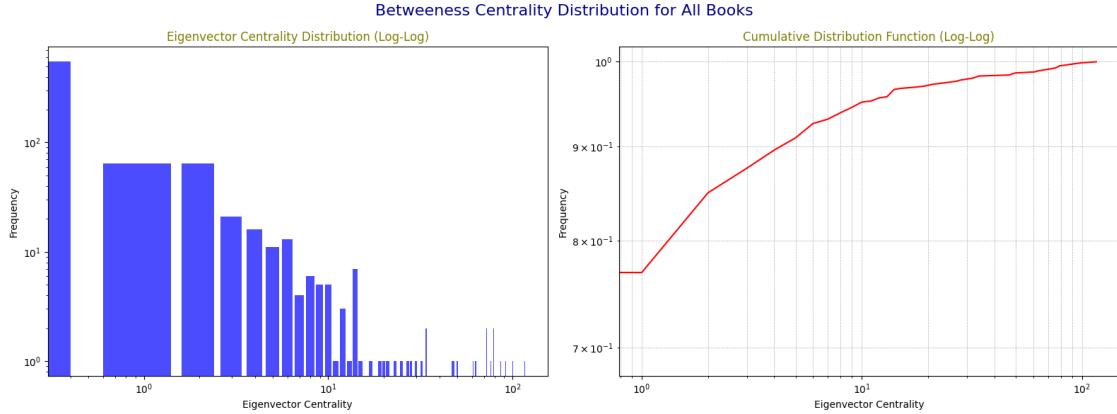
```
[ ]: plot_betweenness_centrality_histogram_and_cdf(G_4, title="Betweenness Centrality Distribution for Book 4")
```



```
[ ]: plot_betweenness_centrality_histogram_and_cdf(G_5, title="Betweenness Centrality Distribution for Book 5")
```



```
[ ]: plot_betweenness_centrality_histogram_and_cdf(G_all, title="Betweeness_Centrality Distribution for All Books")
```



6.2.6 Local Clustering Coefficient

```
[ ]: def clustering_coefficient_VS_degree(G, title=""):
    # Calcolo del degree centrality e del clustering coefficient
    degree_centrality = nx.degree_centrality(G)
    clustering_coeff = nx.clustering(G, nodes=None, weight=None)

    # Raggruppamento dei dati
    degree_values = list(degree_centrality.values())
    degree_values.sort()
    clustering_values = list(clustering_coeff.values())
    clustering_values.sort()

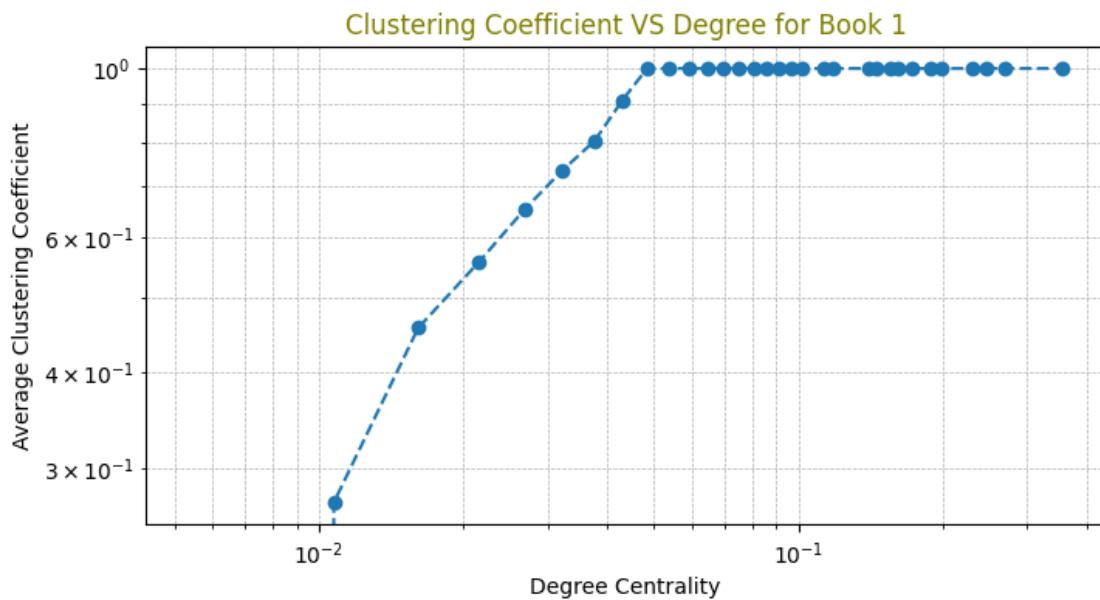
    degree_avg = {}
    for degree, clustering in zip(degree_values, clustering_values):
        if degree not in degree_avg:
            degree_avg[degree] = []
        degree_avg[degree].append(clustering)

    # Calcolo del clustering coefficient medio per ogni degree
    degree_keys = []
    clustering_means = []
    for degree, clustering_list in degree_avg.items():
        degree_keys.append(degree)
        clustering_means.append(np.mean(clustering_list))

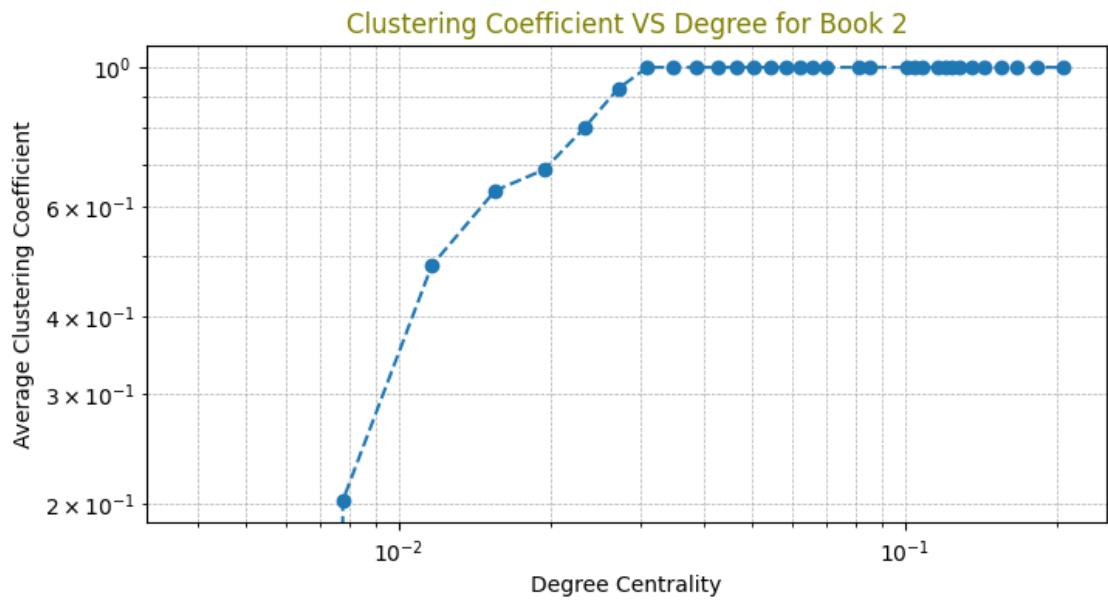
    # Creazione del grafico
```

```
plt.figure(figsize=(8, 4))
plt.loglog(degree_keys, clustering_means, marker='o', linestyle='--')
plt.title(title, color="olive")
plt.xlabel("Degree Centrality")
plt.ylabel("Average Clustering Coefficient")
plt.grid(True, which="both", ls="--", linewidth=0.5)
plt.show()
```

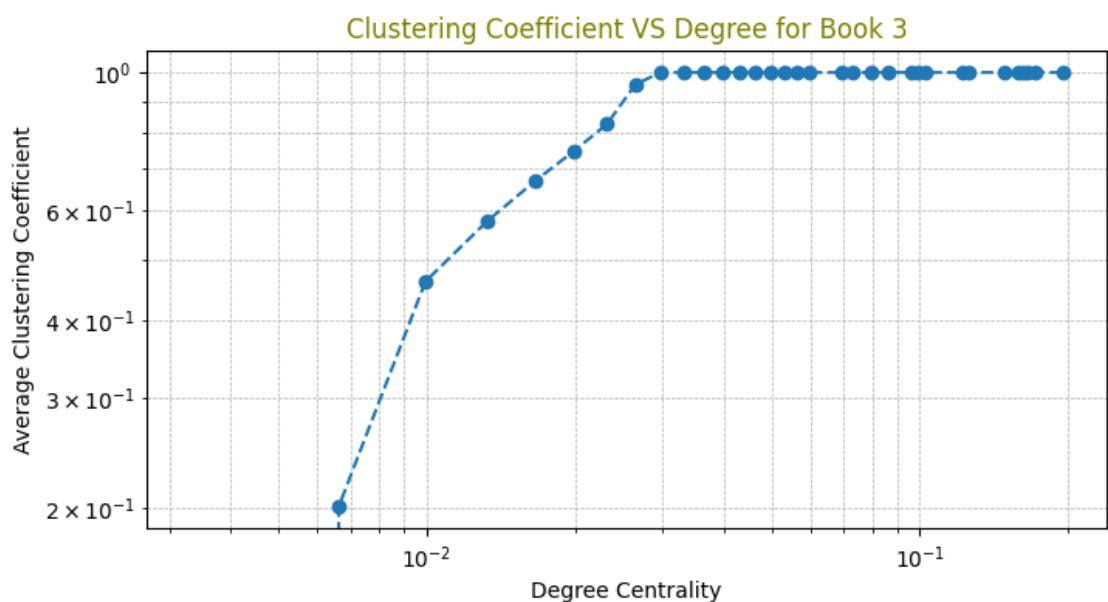
```
[ ]: clustering_coefficient_VS_degree(G_1, title="Clustering Coefficient VS Degree  
for Book 1")
```



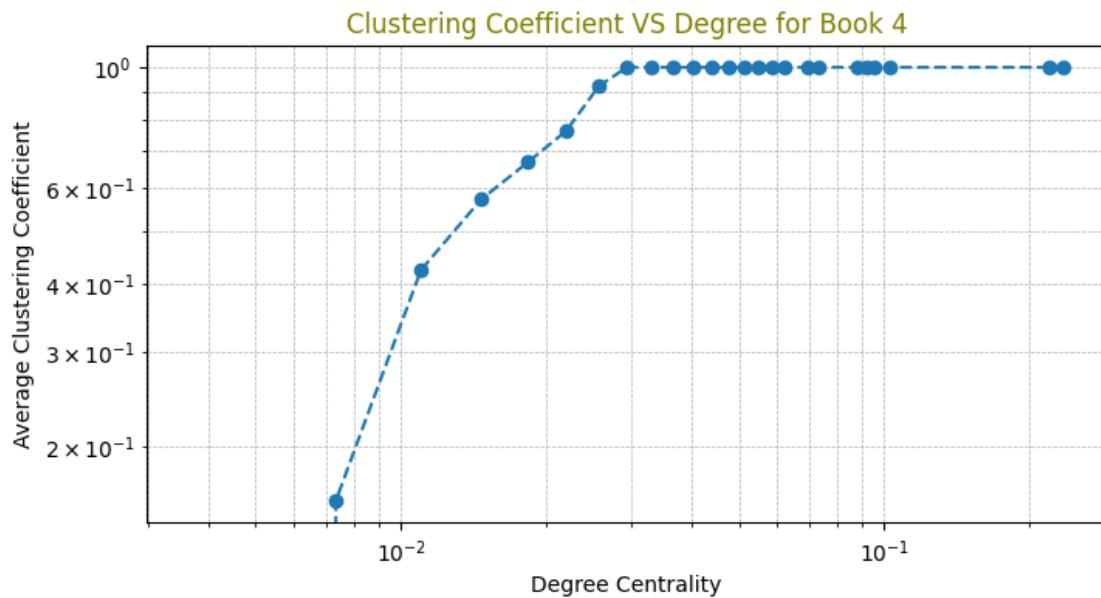
```
[ ]: clustering_coefficient_VS_degree(G_2, title="Clustering Coefficient VS Degree  
for Book 2")
```



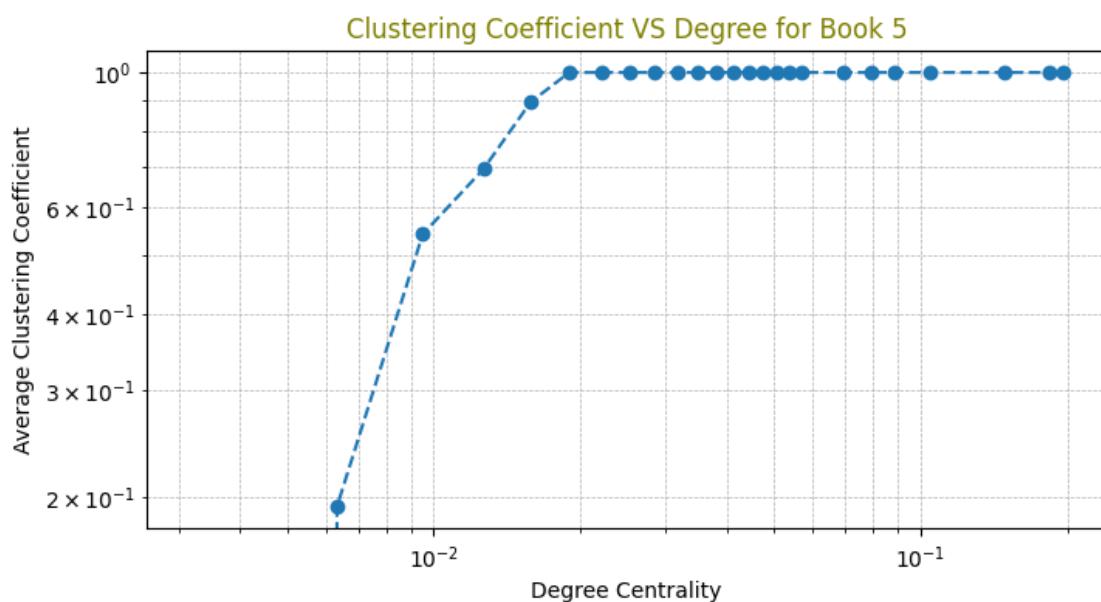
```
[ ]: clustering_coefficient_VS_degree(G_3, title="Clustering Coefficient VS Degree for Book 3")
```



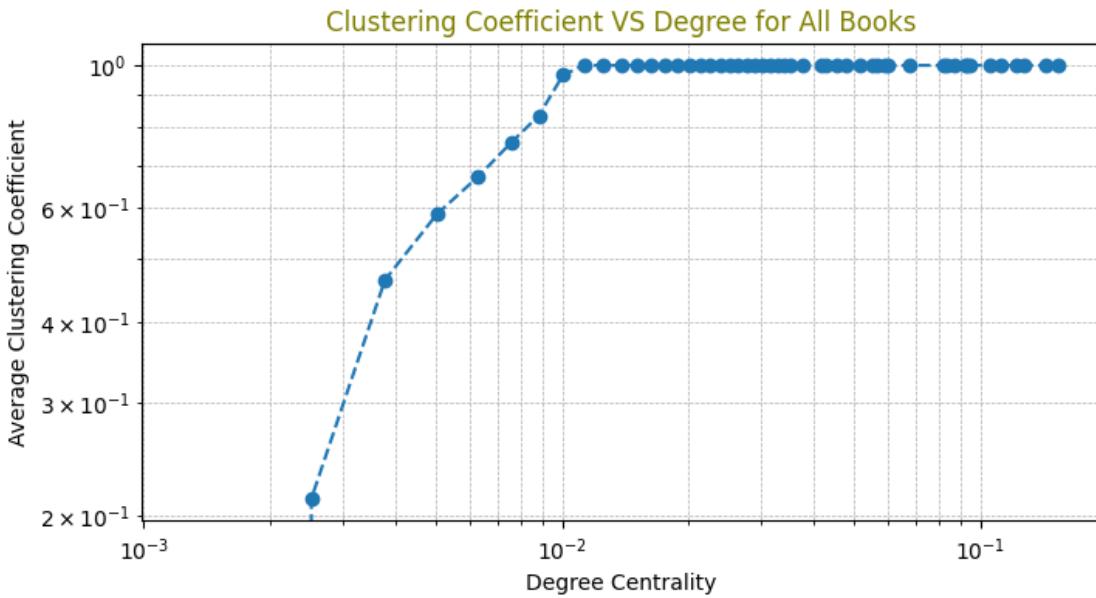
```
[ ]: clustering_coefficient_VS_degree(G_4, title="Clustering Coefficient VS Degree for Book 4")
```



```
[ ]: clustering_coefficient_VS_degree(G_5, title="Clustering Coefficient VS Degree for Book 5")
```



```
[ ]: clustering_coefficient_VS_degree(G_all, title="Clustering Coefficient VS Degree for All Books")
```



Qui noi troviamo che i nodi con molte connessioni (alto grado) tendono ad avere anche molti vicini che sono interconnessi tra loro. Questo suggerisce che i nodi non solo sono popolari, ma anche che formano gruppi coesi. I nodi si raggruppano in comunità fortemente collegate.

ha senso se si pensa al fatto che la trama si divide in zone geograficamente lontane con gruppi di personaggi che sono molto legati tra loro.

6.2.7 Cohesion

Density

```
[ ]: d_1 = nx.density(G_1)
d_2 = nx.density(G_2)
d_3 = nx.density(G_3)
d_4 = nx.density(G_4)
d_5 = nx.density(G_5)
d_all = nx.density(G_all)
```

```
[ ]: print(f"The density for Book 1 is: {d_1:.3f}")
print(f"The density for Book 2 is: {d_2:.3f}")
print(f"The density for Book 3 is: {d_3:.3f}")
print(f"The density for Book 4 is: {d_4:.3f}")
print(f"The density for Book 5 is: {d_5:.3f}")
print(f"The density for all Books is: {d_all:.3f}")
```

The density for Book 1 is: 0.039

The density for Book 2 is: 0.023

The density for Book 3 is: 0.022

The density for Book 4 is: 0.018

```
The density for Book 5 is: 0.015
The density for all Books is: 0.009
```

Il libro 1 è quello con density più alta perchè racconta principalmente le vicende che accadono a Kings Landing e quelle intorno alla famiglia Stark.

Più si va avanti nei libri e più la storia si incasina, quindi la density diminuisce.

Connectedness

```
[ ]: def calculate_connectedness(graph):
    # Numero totale di nodi
    n = graph.number_of_nodes()

    # Trova i componenti connessi
    components = list(nx.connected_components(graph))

    # Inizializza la sommatoria
    connected_sum = 0

    # Calcola  $r_{ij}$  per ogni coppia di nodi
    for component in components:
        size = len(component)
        connected_sum += size * (size - 1)  # Ogni coppia all'interno dello stesso componente

    # Calcola la connectedness
    connectedness = connected_sum / (n * (n - 1))

    return connectedness
```

```
[ ]: conn_1 = calculate_connectedness(G_1)
conn_2 = calculate_connectedness(G_2)
conn_3 = calculate_connectedness(G_3)
conn_4 = calculate_connectedness(G_4)
conn_5 = calculate_connectedness(G_5)
conn_all = calculate_connectedness(G_all)
```

```
[ ]: print(conn_1)
print(conn_2)
print(conn_3)
print(conn_4)
print(conn_5)
print(conn_all)
```

```
1.0
1.0
1.0
```

```
1.0  
1.0  
1.0
```

Dato che siamo di fronte ad un grafo un-directed e che tutti i nodi sono raggiungibili da tutti, non è un grafo disconnesso, allora è ovvio che la connectedness è 1.

Compactness

```
[ ]: def calculate_compactness(graph):  
    # Numero totale di nodi  
    n = graph.number_of_nodes()  
  
    # Calcola le distanze geodetiche  
    distances = dict(nx.all_pairs_shortest_path_length(graph))  
  
    # Inizializza la sommatoria  
    compactness_sum = 0  
  
    # Ottieni la lista dei nodi  
    nodes = list(graph.nodes())  
  
    # Calcola la sommatoria dei reciproci delle distanze  
    for i in range(n):  
        for j in range(i + 1, n):  
            node_i = nodes[i]  
            node_j = nodes[j]  
  
            if node_j in distances[node_i]:  
                distance = distances[node_i][node_j]  
                compactness_sum += 1 / distance # Aggiungi il reciproco della  
            ↪distanza  
  
    # Calcola la compactness  
    compactness = compactness_sum / (n * (n - 1))  
  
    return compactness
```

```
[ ]: comp_1 = calculate_compactness(G_1)  
comp_2 = calculate_compactness(G_2)  
comp_3 = calculate_compactness(G_3)  
comp_4 = calculate_compactness(G_4)  
comp_5 = calculate_compactness(G_5)  
comp_all = calculate_compactness(G_all)
```

```
[ ]: print(f"The compactness in Book 1 is: {comp_1:.3f}")  
print(f"The compactness in Book 2 is: {comp_2:.3f}")
```

```

print(f"The compactness in Book 3 is: {comp_3:.3f}")
print(f"The compactness in Book 4 is: {comp_4:.3f}")
print(f"The compactness in Book 5 is: {comp_5:.3f}")
print(f"The compactness in all Books is: {comp_all:.3f}")

```

The compactness in Book 1 is: 0.193
The compactness in Book 2 is: 0.165
The compactness in Book 3 is: 0.166
The compactness in Book 4 is: 0.148
The compactness in Book 5 is: 0.150
The compactness in all Books is: 0.160

Nessuno dei grafi è particolarmente compatto, questo, ancora una volta, è dovuto dalla natura della trama. Infatti non è detto che tutti i nodi possano facilmente raggiungere gli altri.

Il libro 1 è il più compatto, con i personaggi che principalmente sono a Kings Landing.

6.2.8 Reciprocity

Il nostro grafo è UNDIRECTED, quindi non ha senso calcolare la reciprocità

6.2.9 Transitivity

La transitivity si può calcolare in due modi, da quello che viene detto nelle slides:

- usare la weighted overall clustering coefficient;
- triad census (implementata da nx.transitivity).

Un valore di transitività vicino a 1 indica che i nodi tendono a formare triadi complete, mentre un valore vicino a 0 indica che le connessioni tra i nodi sono più sparse.

```

[ ]: transitivity_1 = nx.transitivity(G_1)
transitivity_2 = nx.transitivity(G_2)
transitivity_3 = nx.transitivity(G_3)
transitivity_4 = nx.transitivity(G_4)
transitivity_5 = nx.transitivity(G_5)
transitivity_all = nx.transitivity(G_all)

```

```

[ ]: print(f"The transitivity value for Book 1 is {transitivity_1:.3f}")
print(f"The transitivity value for Book 2 is {transitivity_2:.3f}")
print(f"The transitivity value for Book 3 is {transitivity_3:.3f}")
print(f"The transitivity value for Book 4 is {transitivity_4:.3f}")
print(f"The transitivity value for Book 5 is {transitivity_5:.3f}")
print(f"The transitivity value for All Books is {transitivity_all:.3f}")

```

The transitivity value for Book 1 is 0.330
The transitivity value for Book 2 is 0.300
The transitivity value for Book 3 is 0.277
The transitivity value for Book 4 is 0.221

```
The transitivity value for Book 5 is 0.203
The transitivity value for All Books is 0.209
```

6.2.10 Centralisation and Core-periphery Indices

Centralization Indices * **Degree centralization:** Measures how much one or a few nodes dominate in terms of number of connections. * **Betweenness centralization:** Captures how much one or a few nodes dominate the flow of information across the network.

Un valore piu' alto di entrambe significa che i nodi presi in considerazione sono piu' centralizzati.

Core-Periphery is computed in two different ways: 1. Degree-based method: separare i nodi in un nucleo molto connesso e una periferia poco connessa tra se', ma potenzialmente collegata al core. Struttura: 1. Ordina i nodi per grado decrescente. 2. Per ogni possibile valore di k (valore di soglia tra nodi core e periphery), si definiscono come i primi k i nodi di core, mentre gli altri come periphery. 3. Si calcola la periphery . 4. Si cerca il valore k che minimizza (cioe' al configurazione di nodi in periferia che sono meno connessi tra loro). 2. K-core decomposition: un k -core e' un sottoinsieme di nodi in cui ogni nodo ha almeno k vicini all'interno del core stesso. Piu' alto e' k e piu' il nucleo e' "resistente" alla rimozione. Struttura: 1. Calcola il core number di ogni nodo, cioe' il massimo valore k per cui il nodo appartiene a un k-core. 2. Si prendono i nodi con core number massimo (quelli con core piu' denso). 3. Si considerano gli altri nodi come periphery.

```
[ ]: def centralisation_indices(G):
    """
    Returns Freeman's degree and betweenness centralisation.
    - Degree: C_D = Σ(max_deg - deg(v)) / ((n-1)(n-2))
    - Between: C_B = Σ(max_bt - bt(v)) / ((n-1)(n-2)/2)
    """
    n = G.number_of_nodes()
    # degree centralisation
    deg = dict(G.degree())
    max_deg = max(deg.values())
    C_deg = sum(max_deg - d for d in deg.values()) / ((n-1)*(n-2))
    # betweenness centralisation (non normalizzata)
    bt = nx.betweenness_centrality(G, normalized=False)
    max_bt = max(bt.values())
    C_bet = sum(max_bt - b for b in bt.values()) / ((n-1)*(n-2)/2)
    return C_deg, C_bet

def core_periphery_degree(G):
    """
    Degree-based dichotomous core-periphery:
    - Ordino nodi per grado decrescente
    - Per ogni possibile k, definisco core = primi k, periphery = resto
    - Calcolo = Σ_{i<j in periphery} (A_ij - p), con p = densità del grafo
    - Scelgo k che MINIMIZZA
    Restituisce (core_set, periphery_set, rho_min).
    """
    nodes = list(G.nodes())
```

```

idx   = {u:i for i,u in enumerate(nodes)}
A     = nx.to_numpy_array(G, nodelist=nodes, dtype=float)
p     = nx.density(G)
# ordina per grado
sorted_nodes = sorted(nodes, key=lambda u: G.degree(u), reverse=True)

best_rho = np.inf
best_k   = 0
for k in range(1, len(nodes)):
    periph = sorted_nodes[k:]
    # sommo solo su i < j
    rho = sum(
        (A[idx[u], idx[v]] - p)
        for i,u in enumerate(periph)
        for v in periph[i+1:])
    )
    if rho < best_rho:
        best_rho, best_k = rho, k

core    = set(sorted_nodes[:best_k])
periphery= set(sorted_nodes[best_k:])
return core, periphery, best_rho

def core_periphery_kcore(G):
    """
    k-core method:
    - Calcolo core-number di ogni nodo
    - Definisco il core come l'insieme dei nodi con core-number massimo
    - Il resto è periphery
    Restituisce (core_set, periphery_set, max_core_number).
    """
    cn = nx.core_number(G)
    max_cn = max(cn.values())
    core    = {u for u,v in cn.items() if v == max_cn}
    periphery= set(G.nodes()) - core
    return core, periphery, max_cn

```

```

[ ]: graphs = {
    "Book I": G_1,
    "Book II": G_2,
    "Book III": G_3,
    "Book IV": G_4,
    "Book V": G_5,
}

for title, G in graphs.items():
    C_deg, C_bet = centralisation_indices(G)

```

```

cd, pd, rho = core_periphery_degree(G)
ck, pk, knum = core_periphery_kcore(G)

print(f"\n== {title} ==")
print(f"Degree centralisation: {C_deg:.4f}")
print(f"Betweenness centralisation:{C_bet:.4f}")
print(f"\n- Core-periphery (degree-based):")
print(f"  • core    ({len(cd)} nodi)")
print(f"  • periphery({len(pd)} nodi)")
print(f"  • minimo = {rho:.4f}")

print(f"\n- Core-periphery (k-core, k={knum}):")
print(f"  • core    ({len(ck)} nodi)")
print(f"  • periphery({len(pk)} nodi)")

```

==== Book I ===

Degree centralisation: 0.3189
 Betweenness centralisation: 48.4828

- Core-periphery (degree-based):
 - core (71 nodi)
 - periphery(116 nodi)
 - minimo = -141.3357
- Core-periphery (k-core, k=11):
 - core (23 nodi)
 - periphery(164 nodi)

==== Book II ===

Degree centralisation: 0.1836
 Betweenness centralisation: 46.2853

- Core-periphery (degree-based):
 - core (90 nodi)
 - periphery(169 nodi)
 - minimo = -142.2898
- Core-periphery (k-core, k=11):
 - core (12 nodi)
 - periphery(247 nodi)

==== Book III ===

Degree centralisation: 0.1745
 Betweenness centralisation: 65.4202

- Core-periphery (degree-based):

- core (123 nodi)
 - periphery(180 nodi)
 - minimo = -162.9249
- Core-periphery (k-core, k=10):
- core (11 nodi)
 - periphery(292 nodi)
- ==== Book IV ====
Degree centralisation: 0.2178
Betweenness centralisation:72.7030
- Core-periphery (degree-based):
- core (137 nodi)
 - periphery(137 nodi)
 - minimo = -99.8755
- Core-periphery (k-core, k=7):
- core (19 nodi)
 - periphery(255 nodi)

- ==== Book V ====
Degree centralisation: 0.1822
Betweenness centralisation:140.7439
- Core-periphery (degree-based):
- core (124 nodi)
 - periphery(193 nodi)
 - minimo = -141.1420
- Core-periphery (k-core, k=6):
- core (17 nodi)
 - periphery(300 nodi)

7 Conclusions and Results