Lab1. # Graph Data Visualization with Python

Winter is Coming...

Game of Thrones is the hugely popular television series based on the equally famous book series `A Song of Ice and Fire` by George R.R.

Let's process & visualize ASAP some data (in different formats) from the **Game of Thrones datasets**.

Task1. Kings battles visualization in Game of Thrones

In this task, we will need to analyze the co-occurrence network of the kings who participated in the same battles in the Game of Thrones.

The *game-of-thrones-battles.csv* dataset stores history of the battles of the War of the Five Kings.

We will build and visualize a directed graph where nodes are kings (attaking kings & defending kings) and the directed edges represent who is being attacked by whom (by participating in the same battle).

```
import pandas as pd
from pyvis.network import Network
```

First, load the *game-of-thrones-battles.csv* file into a Pandas DataFrame

```
#Loading the data
data = pd.read_csv("data/game-of-thrones-battles.csv")
data.head()
```

	name	year	battle_number	attacker_king	defender_king	attacker_1	attacker_2
0	Battle of the Golden Tooth	298	1	Joffrey/Tommen Baratheon	Robb Stark	Lannister	NaN
1	Battle at the Mummer's Ford	298	2	Joffrey/Tommen Baratheon	Robb Stark	Lannister	NaN
2	Battle of Riverrun	298	3	Joffrey/Tommen Baratheon	Robb Stark	Lannister	NaN
3	Battle of the Green Fork	298	4	Robb Stark	Joffrey/Tommen Baratheon	Stark	NaN
4	Battle of the Whispering Wood	298	5	Robb Stark	Joffrey/Tommen Baratheon	Stark	Tully

5 rows × 25 columns

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38 entries, 0 to 37
Data columns (total 25 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   name                38 non-null     object 
 1   year                38 non-null     int64  
 2   battle_number       38 non-null     int64  
 3   attacker_king       36 non-null     object 
 4   defender_king       35 non-null     object 
 5   attacker_1          38 non-null     object 
 6   attacker_2          10 non-null     object 
 7   attacker_3          3 non-null      object 
 8   attacker_4          2 non-null      object 
 9   defender_1          37 non-null     object 
 10  defender_2          2 non-null      object 
 11  defender_3          0 non-null      float64
 12  defender_4          0 non-null      float64
 13  attacker_outcome    37 non-null     object 
 14  battle_type         37 non-null     object 
 15  major_death         37 non-null     float64
 16  major_capture       37 non-null     float64
 17  attacker_size       24 non-null     float64
 18  defender_size       19 non-null     float64
 19  attacker_commander  37 non-null     object 
 20  defender_commander  28 non-null     object 
 21  summer              37 non-null     float64
 22  location            37 non-null     object
 23  region              38 non-null     object
 24  note                5 non-null      object
dtypes: float64(7), int64(2), object(16)
memory usage: 7.6+ KB
```

Select required columns: *'name','attacker_king','defender_king','attacker_size','defender_size'*

```
battles_df=data.loc[:,['name','attacker_king','defender_king','attacker_size','defender_size
battles_df.head()
```

	name	attacker_king	defender_king	attacker_size	defender_size
0	Battle of the Golden Tooth	Joffrey/Tommen Baratheon	Robb Stark	15000.0	4000.0
1	Battle at the Mummer's Ford	Joffrey/Tommen Baratheon	Robb Stark	NaN	120.0
2	Battle of Riverrun	Joffrey/Tommen Baratheon	Robb Stark	15000.0	10000.0
	Battle of the Green		Joffrey/Tommen		

```
battles_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38 entries, 0 to 37
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   name           38 non-null     object
 1   attacker_king  36 non-null     object
 2   defender_king  35 non-null     object
 3   attacker_size  24 non-null     float64
 4   defender_size  19 non-null     float64
dtypes: float64(2), object(3)
memory usage: 1.6+ KB
```

```
#remove rows with any missing values (NaN)
battles_df_cleaned=battles_df.dropna()
battles_df_cleaned.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 16 entries, 0 to 37
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   name           16 non-null     object
 1   attacker_king  16 non-null     object
 2   defender_king  16 non-null     object
 3   attacker_size  16 non-null     float64
 4   defender_size  16 non-null     float64
```

```
dtypes: float64(2), object(3)
memory usage: 768.0+ bytes
```

```
battles_df_cleaned.head(3)
```

	name	attacker_king	defender_king	attacker_size	defender_size
0	Battle of the Golden Tooth	Joffrey/Tommen Baratheon	Robb Stark	15000.0	4000.0
2	Battle of Riverrun	Joffrey/Tommen Baratheon	Robb Stark	15000.0	10000.0

Output names of attacking kings (without repetitions):

```
print(f"Attacking kings: {battles_df_cleaned.attacker_king.unique()}")
```

```
Attacking kings: ['Joffrey/Tommen Baratheon' 'Robb Stark' 'Stannis Baratheon']
```

Output names of defending kings (without repetitions):

Show code

```
Defending kings: ['Robb Stark' 'Joffrey/Tommen Baratheon' 'Balon/Euron Greyjoy'
 'Renly Baratheon' 'Mance Rayder']
```

Instantiate a *Network* object from *pyvis.network*. If working in a Jupyter Notebook environment, set the notebook parameter to True.

https://pyvis.readthedocs.io/en/latest/tutorial.html

```
net5kings = Network(heading="Task1. Building Interactive Network of battles of the War of 5
                bgcolor ="#242020",
                font_color = "white",
                height = "1000px",
                width = "100%",
                directed = True, # we have directed graph
                notebook=True,
                cdn_resources = "remote"

) # do this
```

Define nodes - the list of unique names of all kings. Hint: use Python *set* to avoid repetitions.

Show code

```
Kings list (nodes names): {'Mance Rayder', 'Joffrey/Tommen Baratheon', 'Balon/Euron Grey
```

Add nodes to the graph. Output them via *net5kings.nodes* after that.

Show code

```
Nodes of net5kings properties: [{'color': '#97c2fc', 'id': 'Mance Rayder', 'label': 'Mar
```

Define potential edges as (*king1_name, king2_name*) from *battles_df_cleaned* by using data in ['attacker_king','defender_king'] columns where:

- king1_name - attacking king
- king2_name - defending king

Show code

```
Potential Edges of net5kings:
['Joffrey/Tommen Baratheon', 'Robb Stark']
['Joffrey/Tommen Baratheon', 'Robb Stark']
['Robb Stark', 'Joffrey/Tommen Baratheon']
['Robb Stark', 'Joffrey/Tommen Baratheon']
['Robb Stark', 'Joffrey/Tommen Baratheon']
['Robb Stark', 'Balon/Euron Greyjoy']
['Joffrey/Tommen Baratheon', 'Robb Stark']
['Robb Stark', 'Joffrey/Tommen Baratheon']
['Stannis Baratheon', 'Renly Baratheon']
['Joffrey/Tommen Baratheon', 'Robb Stark']
['Robb Stark', 'Joffrey/Tommen Baratheon']
['Stannis Baratheon', 'Joffrey/Tommen Baratheon']
['Joffrey/Tommen Baratheon', 'Robb Stark']
['Stannis Baratheon', 'Mance Rayder']
['Stannis Baratheon', 'Balon/Euron Greyjoy']
['Stannis Baratheon', 'Joffrey/Tommen Baratheon']
```

Create the list (set) of real edges (without repetitions) based on potential edges defined earlie.

Show code

```
Real (unique) directed Edges of net5kings:
('Stannis Baratheon', 'Mance Rayder')
('Stannis Baratheon', 'Joffrey/Tommen Baratheon')
('Stannis Baratheon', 'Renly Baratheon')
('Robb Stark', 'Balon/Euron Greyjoy')
```

```
('Joffrey/Tommen Baratheon', 'Robb Stark')
('Stannis Baratheon', 'Balon/Euron Greyjoy')
('Robb Stark', 'Joffrey/Tommen Baratheon')
```

Calculate edges weights as the total number of battles between *king1* and *king2*, where *(king1,king2)*- an edge. **Hint**: use groupby with 2 columns and *count()*.

Show code

```
attacker_king             defender_king
Joffrey/Tommen Baratheon  Robb Stark                  5
Robb Stark                Balon/Euron Greyjoy         1
                          Joffrey/Tommen Baratheon    5
Stannis Baratheon         Balon/Euron Greyjoy         1
                          Joffrey/Tommen Baratheon    2
                          Mance Rayder                1
                          Renly Baratheon             1
Name: name, dtype: int64
```

Define the *titles* for edges by using data from *battles_df_cleaned* about battles' *name, attacker_size, and defender_size*.

To join strings within groups in a Pandas DataFrame using *groupby()*, the *agg()* or *apply()* methods can be used in conjunction with the *str.join()* method.

Show code

```
attacker_king             defender_king
Joffrey/Tommen Baratheon  Robb Stark                  Battle of the Golden Tooth,
Battle of Riverrun...
Robb Stark                Balon/Euron Greyjoy                                Battle of
Torrhen's Square
                          Joffrey/Tommen Baratheon    Battle of the Green Fork, Battle
of the Whispe...
Stannis Baratheon         Balon/Euron Greyjoy                                Retaking
of Deepwood Motte
                          Joffrey/Tommen Baratheon        Battle of the Blackwater,
Siege of Winterfell
                          Mance Rayder                                           Battle
of Castle Black
                          Renly Baratheon
Siege of Storm's End
Name: name, dtype: object
```

Assign the weight of each edge and output them as follows:

Show code

```
Attackin king: Stannis Baratheon, Defending king: Mance Rayder, N of battles: 1, battles
Attackin king: Stannis Baratheon, Defending king: Joffrey/Tommen Baratheon, N of battles
Attackin king: Stannis Baratheon, Defending king: Renly Baratheon, N of battles: 1, batt
Attackin king: Robb Stark, Defending king: Balon/Euron Greyjoy, N of battles: 1, battles
Attackin king: Joffrey/Tommen Baratheon, Defending king: Robb Stark, N of battles: 5, ba
Attackin king: Stannis Baratheon, Defending king: Balon/Euron Greyjoy, N of battles: 1,
Attackin king: Robb Stark, Defending king: Joffrey/Tommen Baratheon, N of battles: 5, ba
edges_weights: [1, 2, 1, 1, 5, 1, 5]
```

Add edges with their weigths to the net5kings (via *.add_edge*) and output results as follows:

Show code

```
The edge from Stannis Baratheon to Mance Rayder with weight 1, title: 'Battle of Castle
The edge from Stannis Baratheon to Joffrey/Tommen Baratheon with weight 2, title: 'Battl
The edge from Stannis Baratheon to Renly Baratheon with weight 1, title: 'Siege of Storm
The edge from Robb Stark to Balon/Euron Greyjoy with weight 1, title: 'Battle of Torrher
The edge from Joffrey/Tommen Baratheon to Robb Stark with weight 5, title: 'Battle of th
The edge from Stannis Baratheon to Balon/Euron Greyjoy with weight 1, title: 'Retaking c
The edge from Robb Stark to Joffrey/Tommen Baratheon with weight 5, title: 'Battle of th
```

```
net5kings.edges
```

```
[{'value': 1,
  'title': 'Battle of Castle Black',
  'arrows': 'to',
  'from': 'Stannis Baratheon',
  'to': 'Mance Rayder'},
 {'value': 2,
  'title': 'Battle of the Blackwater, Siege of Winterfell',
  'arrows': 'to',
  'from': 'Stannis Baratheon',
  'to': 'Joffrey/Tommen Baratheon'},
 {'value': 1,
  'title': "Siege of Storm's End",
  'arrows': 'to',
  'from': 'Stannis Baratheon',
  'to': 'Renly Baratheon'},
 {'value': 1,
  'title': "Battle of Torrhen's Square",
  'arrows': 'to',
  'from': 'Robb Stark',
  'to': 'Balon/Euron Greyjoy'},
 {'value': 5,
  'title': 'Battle of the Golden Tooth, Battle of Riverrun, Sack of Winterfell, Battle
of the Fords, The Red Wedding',
  'arrows': 'to',
  'from': 'Joffrey/Tommen Baratheon',
  'to': 'Robb Stark'},
 {'value': 1,
  'title': 'Retaking of Deepwood Motte',
  'arrows': 'to',
  'from': 'Stannis Baratheon',
  'to': 'Balon/Euron Greyjoy'},
 {'value': 5,
  'title': 'Battle of the Green Fork, Battle of the Whispering Wood, Battle of the
Camps, Battle of Oxcross, Sack of Harrenhal',
  'arrows': 'to',
  'from': 'Robb Stark',
  'to': 'Joffrey/Tommen Baratheon'}]
```

Assign the value of node (to scale the node's size) as the N of kings (N+ 1) that this king (a node) has attacked in battles. **Hint**: use *.get_adj_list()* (enemies_map) to retrieve an adjacency list epresentation of the directed graph

Show code

```
{'Mance Rayder': set(),
 'Joffrey/Tommen Baratheon': {'Robb Stark'},
 'Balon/Euron Greyjoy': set(),
 'Robb Stark': {'Balon/Euron Greyjoy', 'Joffrey/Tommen Baratheon'},
 'Stannis Baratheon': {'Balon/Euron Greyjoy',
  'Joffrey/Tommen Baratheon',
  'Mance Rayder',
  'Renly Baratheon'},
 'Renly Baratheon': set()}
```

By using enemies_map (defined earlie) output the following:

Show code

```
King: Mance Rayder has attacked: set(), N of enemies: 0, node's value: 1
King: Joffrey/Tommen Baratheon has attacked: {'Robb Stark'}, N of enemies: 1, node's val
King: Balon/Euron Greyjoy has attacked: set(), N of enemies: 0, node's value: 1
King: Robb Stark has attacked: {'Balon/Euron Greyjoy', 'Joffrey/Tommen Baratheon'}, N of
King: Stannis Baratheon has attacked: {'Balon/Euron Greyjoy', 'Mance Rayder', 'Renly Bar
King: Renly Baratheon has attacked: set(), N of enemies: 0, node's value: 1
```

Use the following color dictionary to assign a *color* to a node according to its *value* (specified earlier):

```
nodeColors={
    0:"blue",
    1: "green",
    2: "orange",
    3: "purple",
    4: "gold",
    5:"red"
}
```

Assign *values* and *color* to nodes. Output results via *net5kings.nodes*

Show code

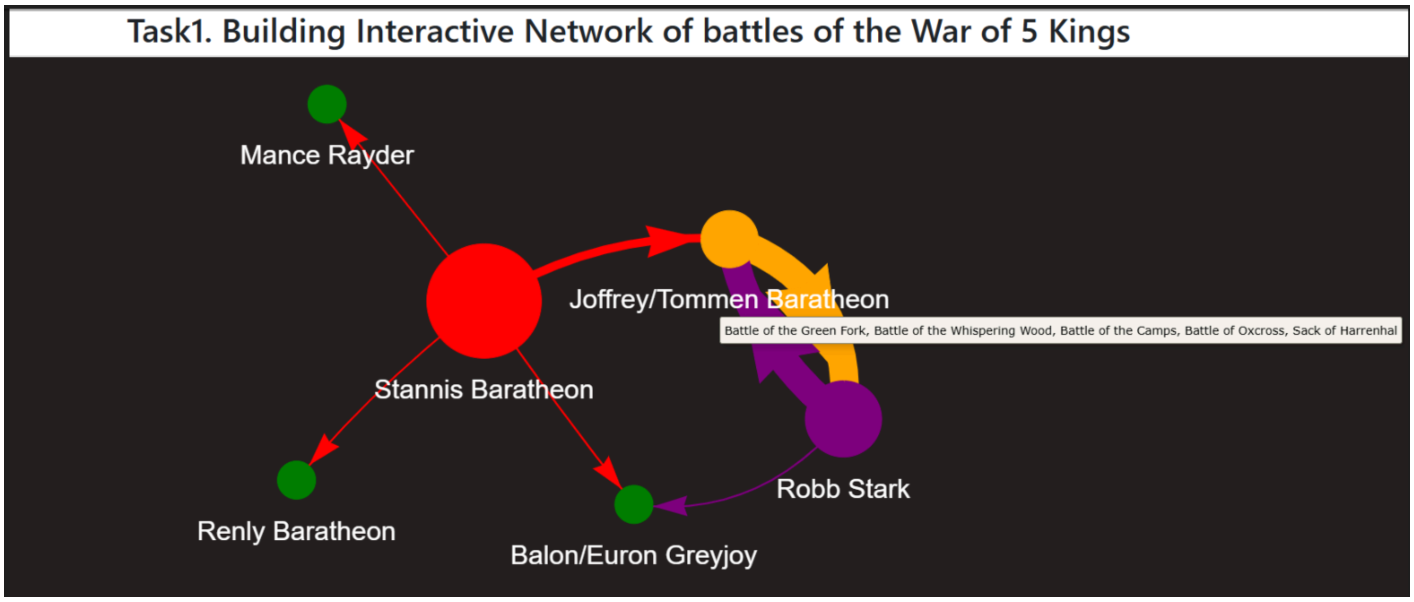
```
net5kings.nodes
```

```
[{'color': 'green',
  'id': 'Mance Rayder',
  'label': 'Mance Rayder',
  'shape': 'dot',
  'font': {'color': 'white'},
  'value': 1},
 {'color': 'orange',
  'id': 'Joffrey/Tommen Baratheon',
  'label': 'Joffrey/Tommen Baratheon',
  'shape': 'dot',
  'font': {'color': 'white'},
  'value': 2},
 {'color': 'green',
  'id': 'Balon/Euron Greyjoy',
  'label': 'Balon/Euron Greyjoy',
  'shape': 'dot',
  'font': {'color': 'white'},
  'value': 1},
 {'color': 'purple',
  'id': 'Robb Stark',
  'label': 'Robb Stark',
  'shape': 'dot',
  'font': {'color': 'white'},
  'value': 3},
 {'color': 'red',
  'id': 'Stannis Baratheon',
  'label': 'Stannis Baratheon',
  'shape': 'dot',
  'font': {'color': 'white'},
  'value': 5},
 {'color': 'green',
  'id': 'Renly Baratheon',
  'label': 'Renly Baratheon',
  'shape': 'dot',
  'font': {'color': 'white'},
  'value': 1}]
```

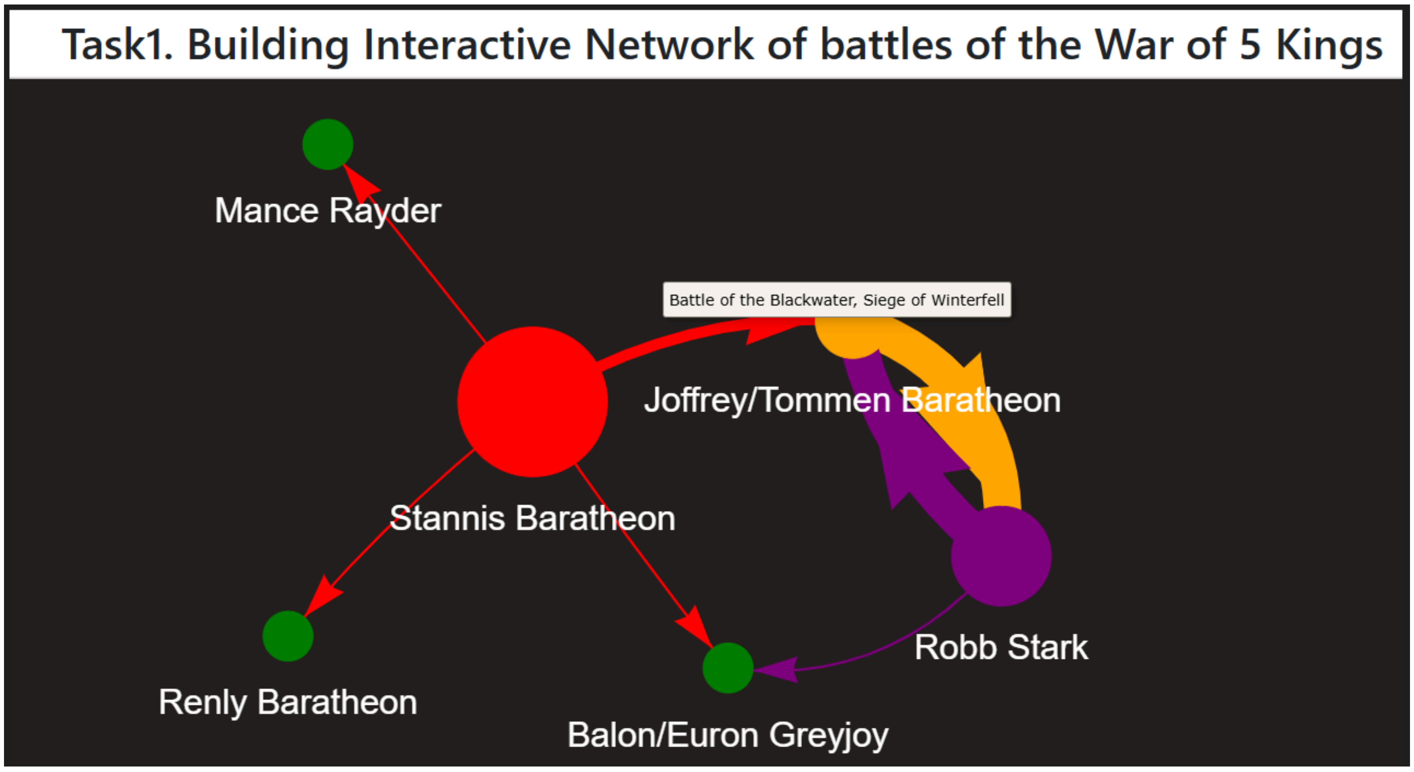
Display the graph

```
net5kings.show("Lab1-task1-net5kings.html", notebook=False)
```

```
Lab1-task1-net5kings.html
```



Battle of the Green Fork, Battle of the Whispering Wood, Battle of the Camps, Battle of Oxcross, Sack of Harrenhal



Battle of the Blackwater, Siege of Winterfell