

## ✓ Manual Implementation of PageRank using BeautifulSoup and Python

```
import requests
from bs4 import BeautifulSoup

# Function to scrape links from a webpage
def scrape_links(url):
    try:
        response = requests.get(url)
        soup = BeautifulSoup(response.content, 'html.parser')
        links = {a['href'] for a in soup.find_all('a', href=True)}
        return links
    except Exception as e:
        print(f"Failed to scrape {url}: {e}")
        return set()

# Websites to analyze
websites = ['https://inc42.com/buzz', 'https://myanimelist.net', 'https://techcrunch.com', 'https://neokyo.com/en/kpop']
```

Start coding or [generate](#) with AI.

```
# Initialize variables for PageRank
d = 0.85 # damping factor
num_iterations = 100
pagerank = {url: 1.0 / len(websites) for url in websites}

# Create a mapping of links
link_map = {url: scrape_links(url) for url in websites}

# Manual PageRank algorithm
for _ in range(num_iterations):
    new_pagerank = {}
    for url in websites:
        new_rank = (1 - d) / len(websites)
        for linking_url in link_map:
            if url in link_map[linking_url]: # If this url links to the current url
                new_rank += d * (pagerank[linking_url] / len(link_map[linking_url]))
        new_pagerank[url] = new_rank
    pagerank = new_pagerank

# Display the PageRank values in sorted order
print("\nPageRank using Manual Implementation (sorted):")
sorted_pagerank = sorted(pagerank.items(), key=lambda x: x[1], reverse=True)
for url, rank in sorted_pagerank:
    print(f"{url}: {rank:.4f}")
```



```
PageRank using Manual Implementation (sorted):
https://neokyo.com/en/kpop: 0.0379
https://techcrunch.com: 0.0377
https://inc42.com/buzz: 0.0375
https://myanimelist.net: 0.0375
```

```
def plot_sorted_pagerank(pagerank):
    sorted_items = sorted(pagerank.items(), key=lambda x: x[1], reverse=True)
    urls, ranks = zip(*sorted_items)

    plt.figure(figsize=(12, 6))
    bars = plt.bar(urls, ranks, color='skyblue')
    plt.xlabel('Websites')
    plt.ylabel('PageRank Score')
    plt.title('Sorted PageRank Scores of Websites')
    plt.xticks(rotation=45)

# Set y-axis limits
plt.ylim(0, max(ranks) + 0.05) # Adjust +0.05 for better visibility

# Add annotations
```

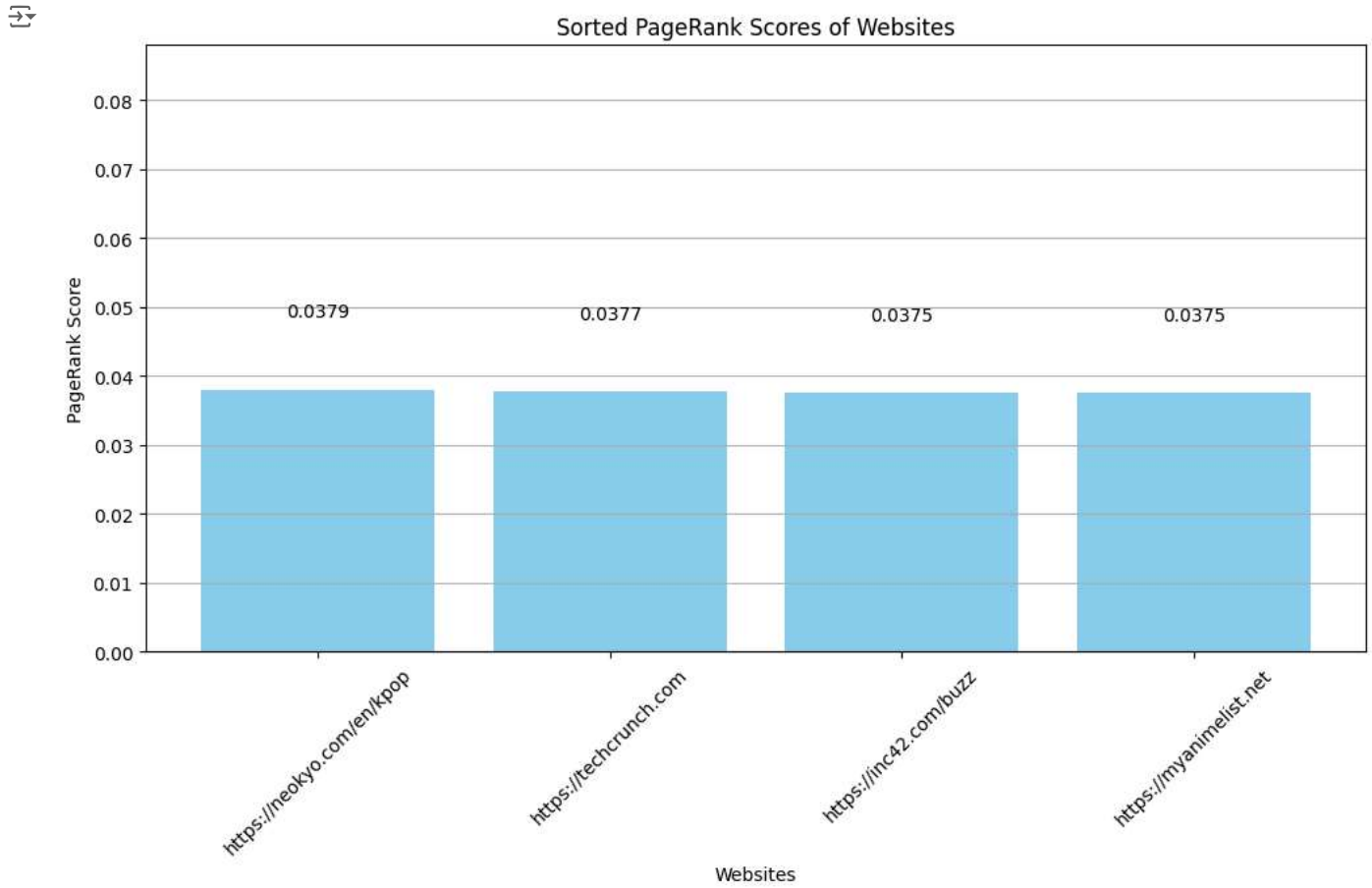
```

for bar in bars:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval + 0.01, f"{yval:.4f}", ha='center', va='bottom')

plt.grid(axis='y') # Add horizontal grid lines
plt.show()

# Plot sorted PageRank scores
plot_sorted_pagerank(pagerank)

```



Start coding or [generate](#) with AI.