Numerical optimization and large scale linear algebra

Stelios Giagkos f3352410

Pagerank

In the file Stanweb.dat, you will find in compressed form the connectivity matrix for the webpages of Stanford University. Specifically in the first column are contained the nodes while in the second the node with which is connected. Using the notation of the tutorial pagerank.pdf do the following:

Section A

a) Find the vector π with:

i) The Power Method

ii) Solving the Corresponding System

As described in paragraphs 5.1 and 5.2 of the tutorial, for both methods consider:

- $ \alpha = 0.85 $
- Stopping criterion $\tau = 10^{-8} $
- The vector a having 1 if it corresponds to a node with no out-links, and 0 otherwise.

Questions:

1. Are the results the same for both methods?
2. Which method seems to be faster?

Use the **Gauss-Seidel method** for the iterative solution of the system.

Imports

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content
/drive", force_remount=True).
```

	Node	Link	Prob
0	1	6548	0.500000
1	1	15409	0.500000
2	2	252915	0.032258
3	2	246897	0.032258
4	2	251658	0.032258
...	...	...	...
2382907	281903	216688	0.142857
2382908	281903	90591	0.142857
2382909	281903	94440	0.142857
2382910	281903	56088	0.142857
2382911	281903	44103	0.142857

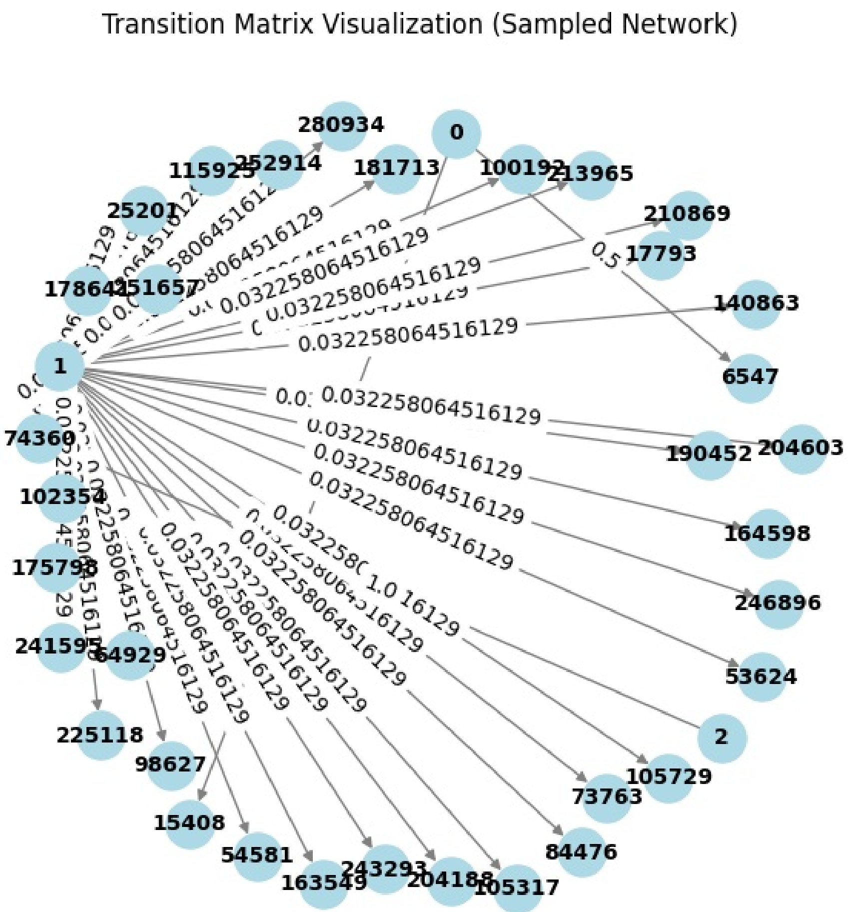
2382912 rows × 3 columns

```
CSR Matrix for node 0:
 <Compressed Sparse Row sparse matrix of dtype 'float64'
	with 2 stored elements and shape (1, 281903)>
  Coords	Values
  (0, 6547)	0.5
  (0, 15408)	0.5
```

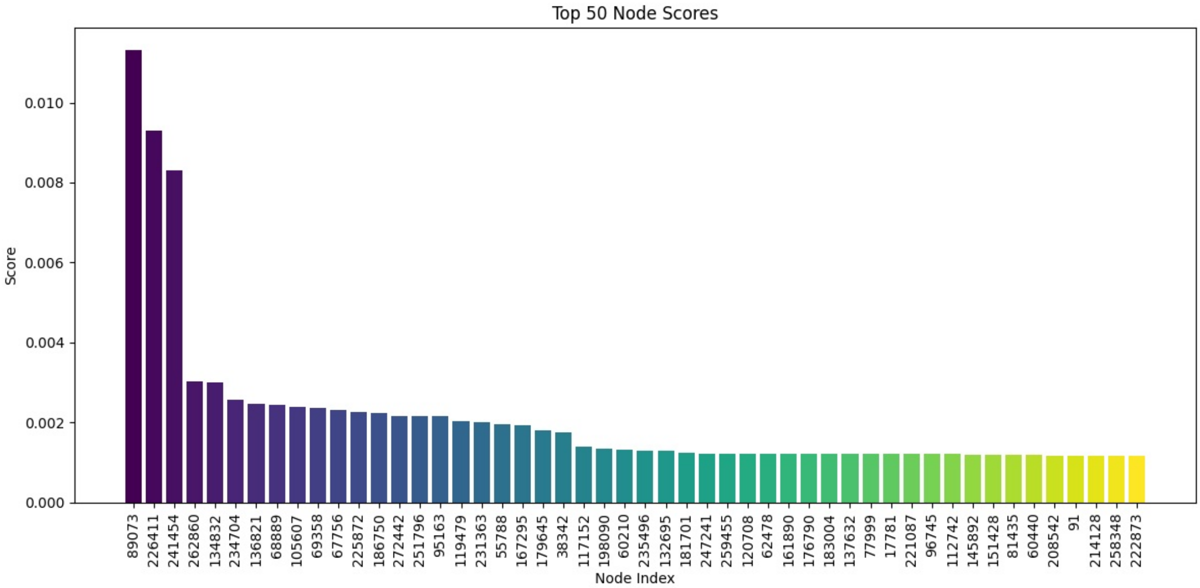
This code visualizes a sampled subgraph from a larger network represented by a transition matrix $ P $. It first selects a small subset of nodes from the matrix to create a smaller, more manageable graph. Then, it constructs a directed graph using NetworkX by adding edges between nodes based on non-zero transition probabilities (weights) in the matrix.

The visualization is created using a spring layout for better aesthetics, and edge weights (transition probabilities) are labeled on the graph. This allows for a clear representation of how the sampled nodes are connected and how the transition probabilities affect the connections between them.

The code is useful for understanding the structure of a Markov chain or similar systems by providing a graphical representation of the relationships between nodes in a sampled portion of the network.



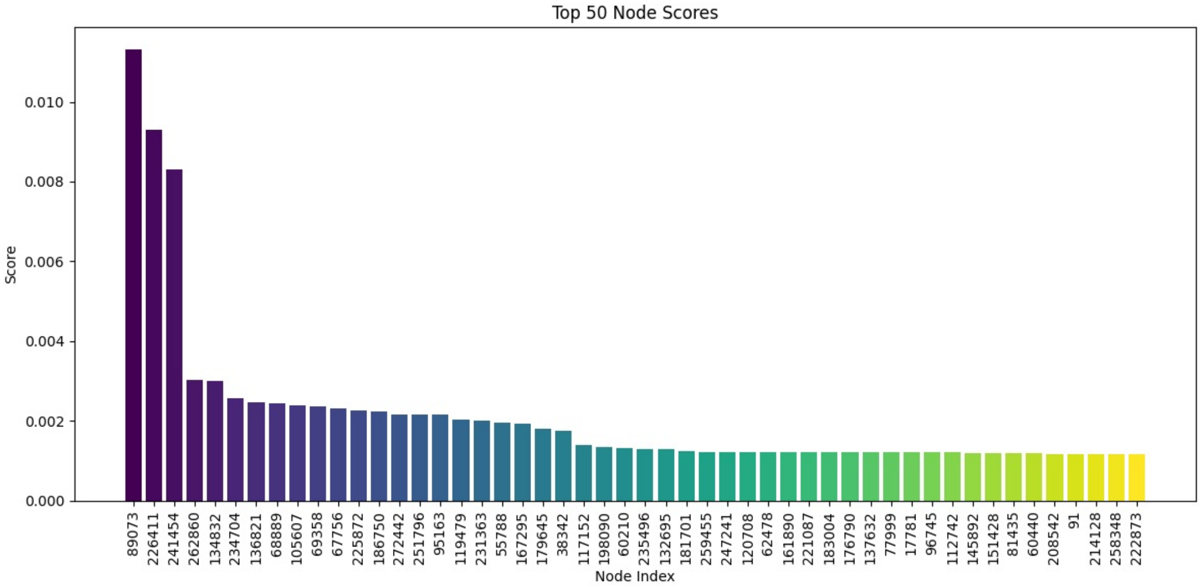
```
Test # 1: Power method converged after 91 iterations (damping factor: 0.85).
Time: 6.7736 seconds
Test # 2: Power method converged after 91 iterations (damping factor: 0.85).
Time: 6.2887 seconds
Test # 3: Power method converged after 91 iterations (damping factor: 0.85).
Time: 5.5503 seconds
Test # 4: Power method converged after 91 iterations (damping factor: 0.85).
Time: 7.5242 seconds
Test # 5: Power method converged after 91 iterations (damping factor: 0.85).
Time: 5.6049 seconds

The average Total time for Power Method with a=0.85 is: 6.3484 seconds.


Power method converged after 91 iterations (damping factor: 0.85).
[ 89073 226411 241454 262860 134832 234704 136821  68889 105607  69358
  67756 225872 186750 272442 251796  95163 119479 231363  55788 167295
 179645  38342 117152 198090  60210 235496 132695 181701 247241 259455
 120708  62478 161890 176790 183004 137632  77999  17781 221087  96745
 112742 145892 151428  81435  60440 208542     91 214128 258348 222873]
```



ii) Gauss-Seidel Method Explanation

This function solves a system of linear equations (Ax = b) iteratively using the Gauss-Seidel method. It starts with an initial guess and repeatedly updates the solution vector until it converges or reaches a maximum number of iterations. The error is tracked if requested. The function returns the approximate solution, the number of iterations, and the error history.

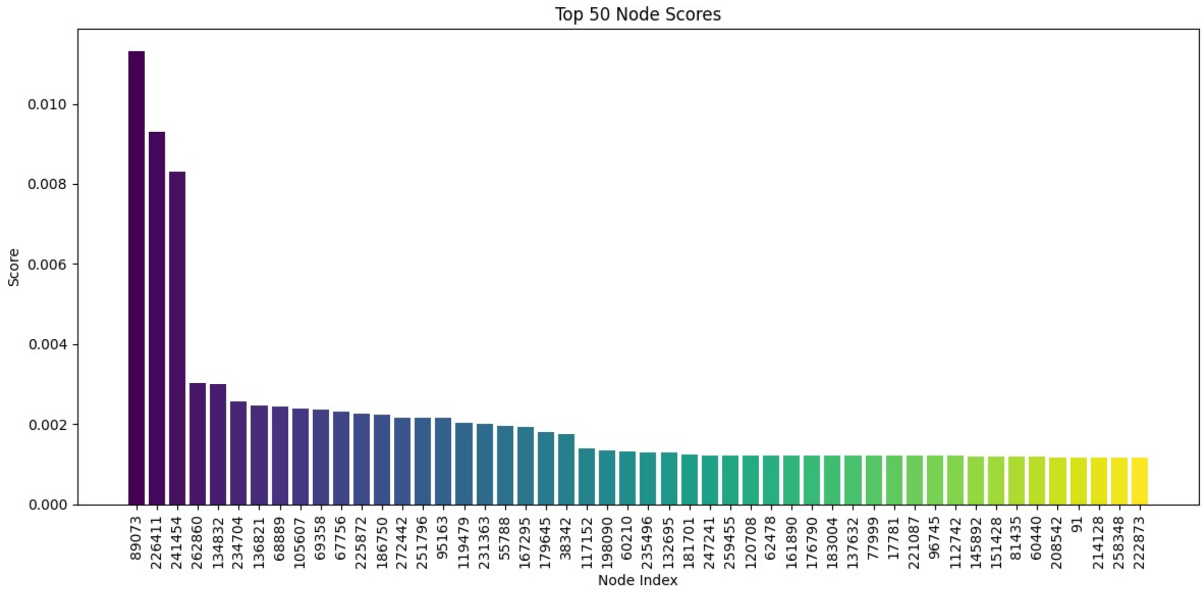
```
Test # 1: Gauss Seidel Method converged after 62 iterations for a=0.85
Time: 5.8294 seconds
Test # 2: Gauss Seidel Method converged after 62 iterations for a=0.85
Time: 6.9892 seconds
Test # 3: Gauss Seidel Method converged after 62 iterations for a=0.85
Time: 5.7790 seconds
Test # 4: Gauss Seidel Method converged after 62 iterations for a=0.85
Time: 6.7199 seconds
Test # 5: Gauss Seidel Method converged after 62 iterations for a=0.85
Time: 5.9742 seconds

The average Total time for System Solver with a=0.85 is: 6.2583 seconds.

Gauss Seidel Method converged after 62 iterations for a=0.85
[ 89073 226411 241454 262860 134832 234704 136821  68889 105607  69358
  67756 225872 186750 272442 251796  95163 119479 231363  55788 167295
 179645  38342 117152 198090  60210 235496 132695 181701 259455 247241
 120708  62478 161890 221087 183004 176790 137632  77999  17781  96745
 112742 145892 151428  81435  60440 208542     91 214128 258348 222873]
```



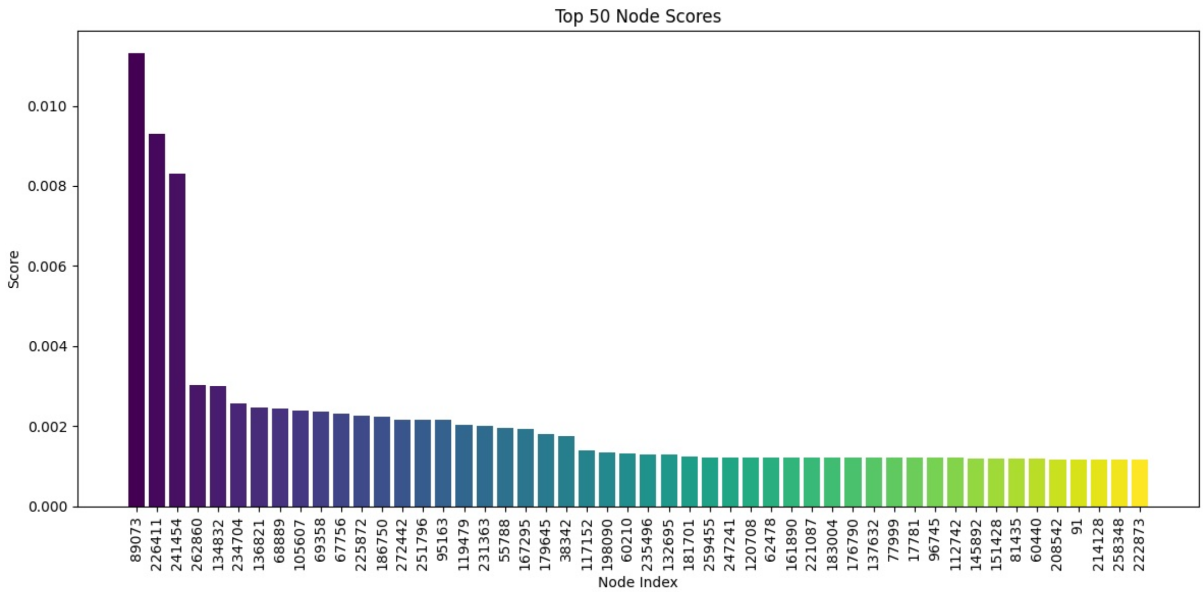

```
Power method converged after 91 iterations (damping factor: 0.85).
Gauss Seidel Method converged after 62 iterations for a=0.85
```

	Power Method Probs	Gauss Seidel Prob
89073	0.011303	0.011303
226411	0.009288	0.009288
241454	0.008297	0.008297
262860	0.003023	0.003023
134832	0.003001	0.003001
234704	0.002572	0.002572
136821	0.002454	0.002454
68889	0.002431	0.002431
105607	0.002397	0.002397
69358	0.002364	0.002364

```
Total number of index differences in the top 50 rankings for Power method (a=0.85)
[ 89073 226411 241454 262860 134832 234704 136821  68889 105607  69358
  67756 225872 186750 272442 251796  95163 119479 231363  55788 167295
 179645  38342 117152 198090  60210 235496 132695 181701 247241 259455
 120708  62478 161890 176790 183004 137632  77999  17781 221087  96745
 112742 145892 151428  81435  60440 208542     91 214128 258348 222873]
```




```
Total number of index differences in the top 50 rankings for Gauss Seidel method (a=0.85)
[ 89073 226411 241454 262860 134832 234704 136821  68889 105607  69358
  67756 225872 186750 272442 251796  95163 119479 231363  55788 167295
 179645  38342 117152 198090  60210 235496 132695 181701 259455 247241
 120708  62478 161890 221087 183004 176790 137632  77999  17781  96745
 112742 145892 151428  81435  60440 208542     91 214128 258348 222873]
```



```
Power vs Gauss Seidel Methods for α=0.85 :

Total number of index differences in the top 50 rankings for α=0.85: 7
Differences DataFrame:
```

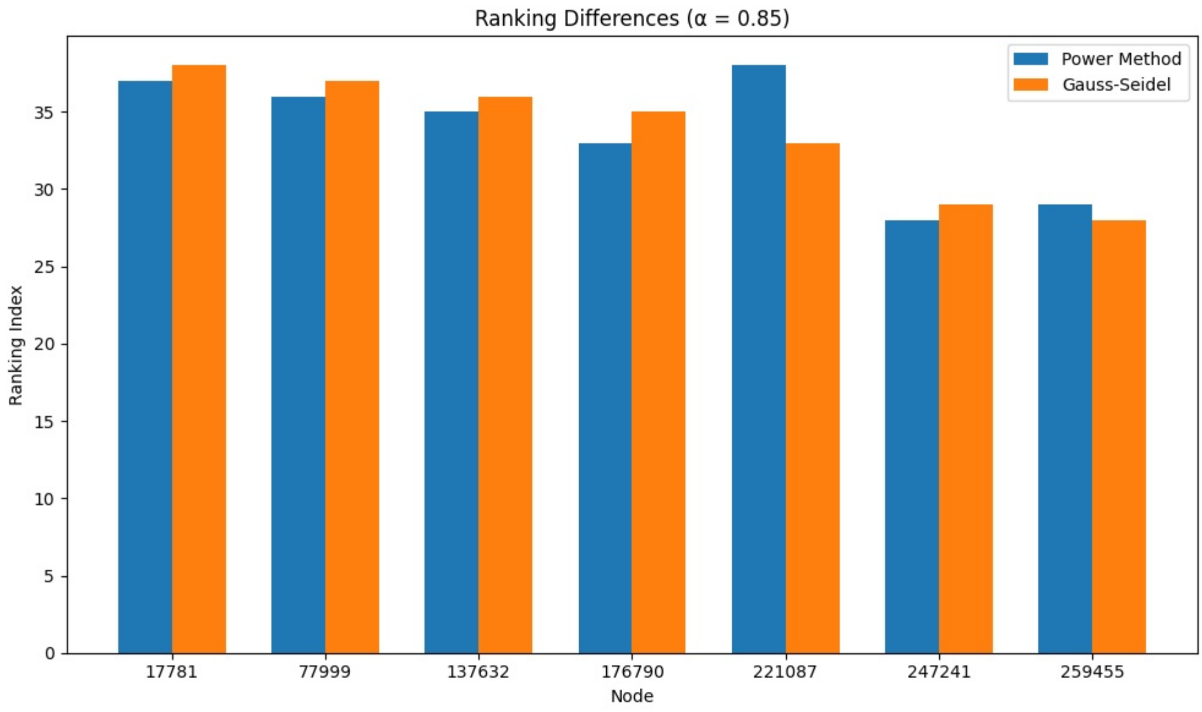
	Node	Power Method Index	Gauss-Seidel Index
0	17781	37	38
1	77999	36	37
2	137632	35	36
3	176790	33	35
4	221087	38	33
5	247241	28	29
6	259455	29	28

Summary of Ranking Changes Between Power Method and Gauss-Seidel Method for α = 0.85

- **Total Number of Index Differences in Top 50 Rankings:**
  - There are **7** differences in the rankings within the top 50 rankings for **α = 0.85**.
- **Differences in Rankings:**
  - **Node 0**: Power Method ranks it **37th**, Gauss-Seidel ranks it **38th** (difference of 1).
  - **Node 1**: Power Method ranks it **36th**, Gauss-Seidel ranks it **37th** (difference of 1).
  - **Node 2**: Power Method ranks it **35th**, Gauss-Seidel ranks it **36th** (difference of 1).
  - **Node 3**: Power Method ranks it **33rd**, Gauss-Seidel ranks it **35th** (difference of 2).
  - **Node 4**: Power Method ranks it **38th**, Gauss-Seidel ranks it **33rd** (difference of 5).
  - **Node 5**: Power Method ranks it **28th**, Gauss-Seidel ranks it **29th** (difference of 1).
  - **Node 6**: Power Method ranks it **29th**, Gauss-Seidel ranks it **28th** (difference of 1).

Conclusion:

The rankings for the nodes in the top 50 rankings under **α = 0.85** are fairly close between the Power Method and Gauss-Seidel method, with **7 total differences**. While the shifts are generally small, **Node 4** exhibits a more significant difference, with a **5-position** shift between the two methods. This suggests that the rankings between the methods are largely similar, but there are slight variations in how each method prioritizes certain nodes.



Comparison of PageRank Methods: Power Method vs. System Solver (Gauss-Seidel)

Results:

Are the results the same for both methods?

Yes, the top 50 node indices were nearly identical between the Power Method and the Gauss-Seidel System Solver. Minor discrepancies (7 nodes) were observed, indicating that both methods effectively converged to a very similar PageRank solution.

Which method seems to be faster?

Power Method:

- Converged in 91 iterations across all tests.
- Average runtime: **6.3484 seconds**.

System Solver (Gauss-Seidel):

- Converged in 62 iterations across all tests.
- Average runtime: **6.2583 seconds**.

Conclusion:

Based on the average runtime, the System Solver (Gauss-Seidel) (6.2583 seconds) was slightly faster than the **Power Method (6.3484 seconds). This is consistent with the observation that the Gauss-Seidel method required fewer iterations (62 vs. 91) to converge.

Key Observations:

- Both methods produce comparable PageRank results.
- The Power Method demonstrates a slight performance advantage in terms of execution time.
- The Gauss-Seidel method required significantly less iterations to converge.

b) Task with $α = 0.99$

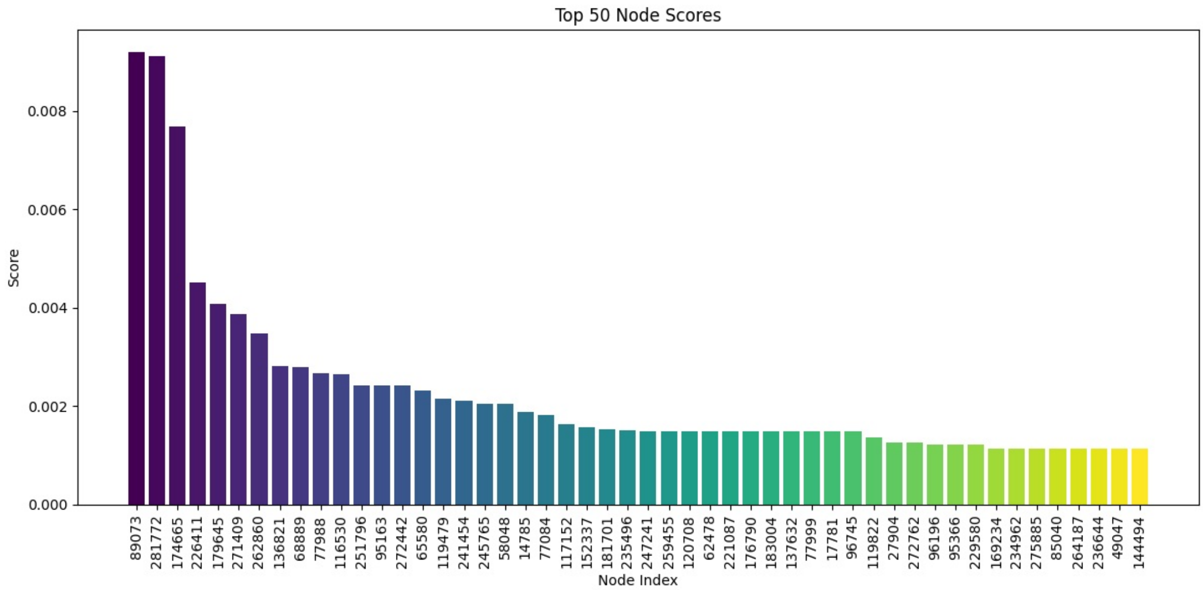
- **Objective**: Perform the previous task with a damping factor of $\alpha = 0.99 $.
- **Remarks**: Discuss the convergence speed of the algorithm.
- **Question**: Did the ranking of the first 50 nodes change compared to the previous result?

Runnign Power Method for $a=0.99$

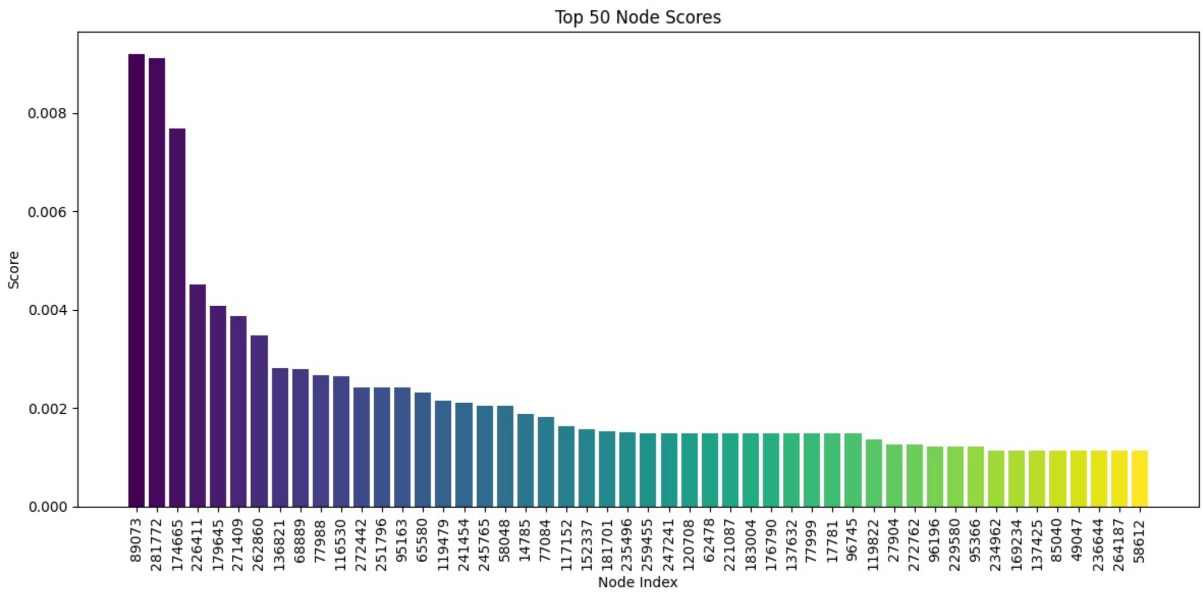
```
Test # 1: Power method converged after 1392 iterations (damping factor: 0.99).
Time: 106.3816 seconds
Test # 2: Power method converged after 1392 iterations (damping factor: 0.99).
Time: 109.1944 seconds
Test # 3: Power method converged after 1392 iterations (damping factor: 0.99).
Time: 121.2876 seconds
Test # 4: Power method converged after 1392 iterations (damping factor: 0.99).
Time: 100.8377 seconds
Test # 5: Power method converged after 1392 iterations (damping factor: 0.99).
Time: 102.7070 seconds

The average Total time for Power Method with a=0.99 is: 108.0817 seconds.

Test # 1: Gauss Seidel Method converged after 968 iterations for a=0.99
Time: 86.8214 seconds
Test # 2: Gauss Seidel Method converged after 968 iterations for a=0.99
Time: 77.5640 seconds
Test # 3: Gauss Seidel Method converged after 968 iterations for a=0.99
Time: 78.4471 seconds
Test # 4: Gauss Seidel Method converged after 968 iterations for a=0.99
Time: 78.0614 seconds
Test # 5: Gauss Seidel Method converged after 968 iterations for a=0.99
Time: 76.9548 seconds

The average Total time for System Solver with a=0.99 is: 79.5697 seconds.

Power method converged after 1392 iterations (damping factor: 0.99).
Gauss Seidel Method converged after 968 iterations for a=0.99
```

	Power Method Probs	Gauss Seidel Prob
89073	0.009187	0.009187
281772	0.009112	0.009112
174665	0.007689	0.007689
226411	0.004514	0.004514
179645	0.004073	0.004073
271409	0.003872	0.003872
262860	0.003485	0.003485
136821	0.002821	0.002821
68889	0.002790	0.002790
77988	0.002676	0.002676

```
Total number of index differences in the top 50 rankings for Power method (a=0.99)
[ 89073 281772 174665 226411 179645 271409 262860 136821  68889  77988
 116530 251796  95163 272442  65580 119479 241454 245765  58048  14785
  77084 117152 152337 181701 235496 247241 259455 120708  62478 221087
 176790 183004 137632  77999  17781  96745 119822  27904 272762  96196
  95366 229580 169234 234962 275885  85040 264187 236644  49047 144494]
```




```
Total number of index differences in the top 50 rankings for Gauss Seidel method (a=0.99)
[ 89073 281772 174665 226411 179645 271409 262860 136821  68889  77988
 116530 272442 251796  95163  65580 119479 241454 245765  58048  14785
  77084 117152 152337 181701 235496 259455 247241 120708  62478 221087
 183004 176790 137632  77999  17781  96745 119822  27904 272762  96196
 229580  95366 234962 169234 137425  85040  49047 236644 264187  58612]
```

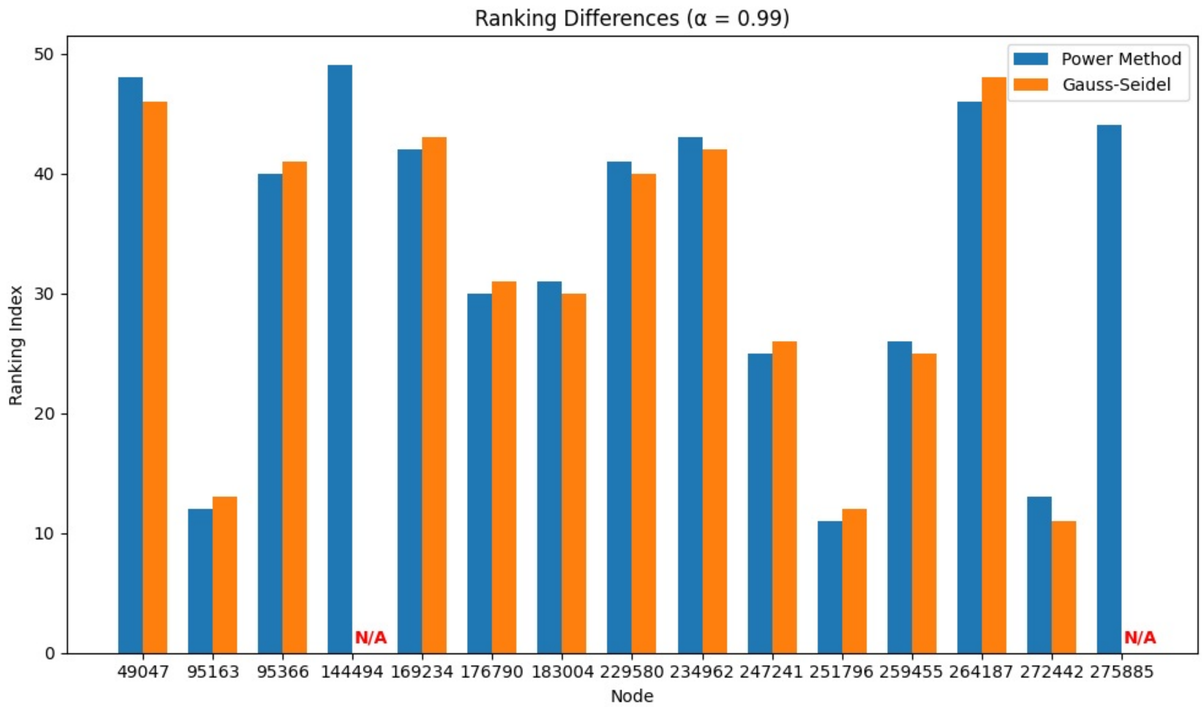



```
Power vs Gauss Seidel Methods for α=0.99 :

Total number of index differences in the top 50 rankings for α=0.99: 15
Differences DataFrame:
```

	Node	Power Method Index	Gauss-Seidel Index
0	49047	48	46
1	95163	12	13
2	95366	40	41
3	144494	49	N/A
4	169234	42	43
5	176790	30	31
6	183004	31	30
7	229580	41	40
8	234962	43	42
9	247241	25	26
10	251796	11	12
11	259455	26	25
12	264187	46	48
13	272442	13	11
14	275885	44	N/A

Summary of Ranking Changes Between Power Method and Gauss-Seidel Method

- **Matching Rankings:**
  - Nodes 1, 2, 4, 5, 6, 7, 8, and 10 show very similar ranks in both methods with minor shifts (e.g., Node 1: 12th in Power Method, 13th in Gauss-Seidel).
- **Larger Differences in Rankings:**
  - **Node 0**: Power Method ranks 49047th, while Gauss-Seidel ranks it 48th, showing a drastic difference.
  - **Node 3**: No ranking in Gauss-Seidel ("N/A"), but ranks 49th in the Power Method, indicating an issue or absence in the Gauss-Seidel computation.
  - **Node 9**: Ranks 25th in Power Method and 26th in Gauss-Seidel, showing a small shift.
  - **Node 13**: Ranks 13th in Power Method and 11th in Gauss-Seidel, showing a slight difference in importance.
- **Nodes with "N/A" in Gauss-Seidel Method:**
  - **Node 3** and **Node 14** are not ranked in the Gauss-Seidel method ("N/A"), indicating either non-convergence .




```
Power  Method for α=0.85 vs Power  Method for α=0.99:

Total number of index differences in the top 50 rankings for the two Power Methods: 48
Differences DataFrame:
```

	Node	Power Method Index	Gauss-Seidel Index
0	91	46	N/A
1	17781	37	34
2	38342	21	N/A
3	55788	18	N/A
4	60210	24	N/A
5	60440	44	N/A
6	62478	31	28
7	67756	10	N/A
8	68889	7	8
9	69358	9	N/A
10	77999	36	33
11	81435	43	N/A
12	95163	15	12
13	96745	39	35
14	105607	8	N/A
15	112742	40	N/A
16	117152	22	21
17	119479	16	15
18	120708	30	27
19	132695	26	N/A
20	134832	4	N/A
21	136821	6	7
22	137632	35	32

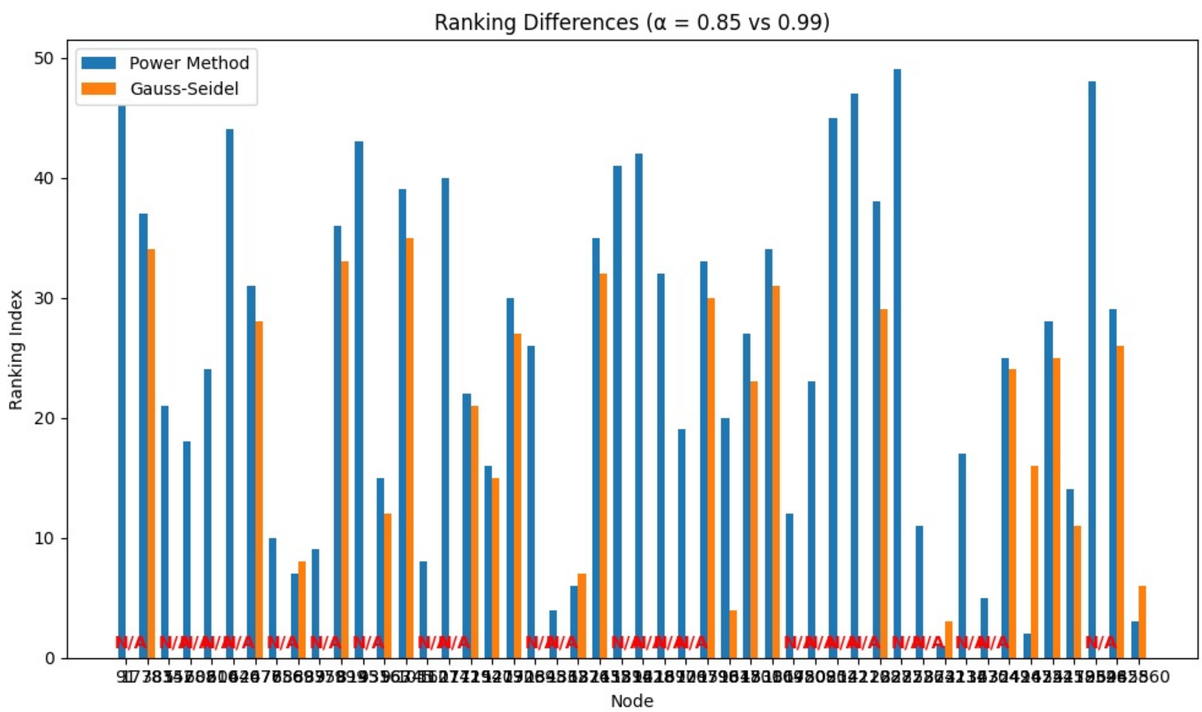
23	145892	41	N/A
24	151428	42	N/A
25	161890	32	N/A
26	167295	19	N/A
27	176790	33	30
28	179645	20	4
29	181701	27	23
30	183004	34	31
31	186750	12	N/A
32	198090	23	N/A
33	208542	45	N/A
34	214128	47	N/A
35	221087	38	29
36	222873	49	N/A
37	225872	11	N/A
38	226411	1	3
39	231363	17	N/A
40	234704	5	N/A
41	235496	25	24
42	241454	2	16
43	247241	28	25
44	251796	14	11
45	258348	48	N/A
46	259455	29	26
47	262860	3	6

Summary of Ranking Changes Between Power Method for α = 0.85 and Power Method for α = 0.99

- **Total Number of Index Differences in the Top 50 Rankings:**
  - There are **48 total differences** between the two Power Methods in the top 50 rankings.
- **General Observations:**
  - The rankings between **Power Method for α = 0.85** and **Power Method for α = 0.99** vary significantly for several nodes, with the largest differences observed in positions like Node 0, Node 2, and Node 4. For example:
    - **Node 0**: Ranked **91st** in α = 0.85 and **46th** in α = 0.99.
    - **Node 2**: Ranked **38342nd** in α = 0.85 and **21st** in α = 0.99.
    - **Node 4**: Ranked **60210th** in α = 0.85 and **24th** in α = 0.99.
- **Nodes with Significant Shifts in Ranking:**
  - Some nodes exhibit very large shifts in ranking, like:
    - **Node 1**: Ranked **17781st** in α = 0.85 and **37th** in α = 0.99 (a change of over 17,000 places).
    - **Node 7**: Ranked **67756th** in α = 0.85 and **10th** in α = 0.99 (a change of over 67,000 places).
    - **Node 14**: Ranked **105607th** in α = 0.85 and **8th** in α = 0.99.

Nodes with "N/A" in Power Method for α = 0.99

- Several nodes did not have a ranking in the Power Method for **α = 0.99**, marked as "N/A".
- These nodes include **Node 0**, **Node 2**, **Node 4**, **Node 5**, **Node 7**, and several others.
- The "N/A" ranking indicates that these nodes were either not considered or failed to converge in the **α = 0.99** method.
- **Nodes with Minimal Changes:**
  - There are several nodes with minimal changes in ranking, such as:
    - **Node 8**: Ranked **68889th** in α = 0.85 and **7th** in α = 0.99 (a small shift).
    - **Node 12**: Ranked **95163rd** in α = 0.85 and **15th** in α = 0.99.
    - **Node 42**: Ranked **241454th** in α = 0.85 and **2nd** in α = 0.99.
- **Conclusion:**
  - The rankings for the two Power Methods, **α = 0.85** and **α = 0.99**, show significant changes, with **48 total differences** in the top 50 nodes. Many nodes experience large shifts in their rankings, while a few show only minimal changes. Furthermore, several nodes were not ranked or did not converge in the **α = 0.99** method ("N/A"). This suggests that the Power Method's performance and ranking priorities are highly sensitive to the choice of **α**, with some nodes being completely excluded from rankings in the higher **α** case.



Power Method for α=0.85 vs Power Method for α=0.99:

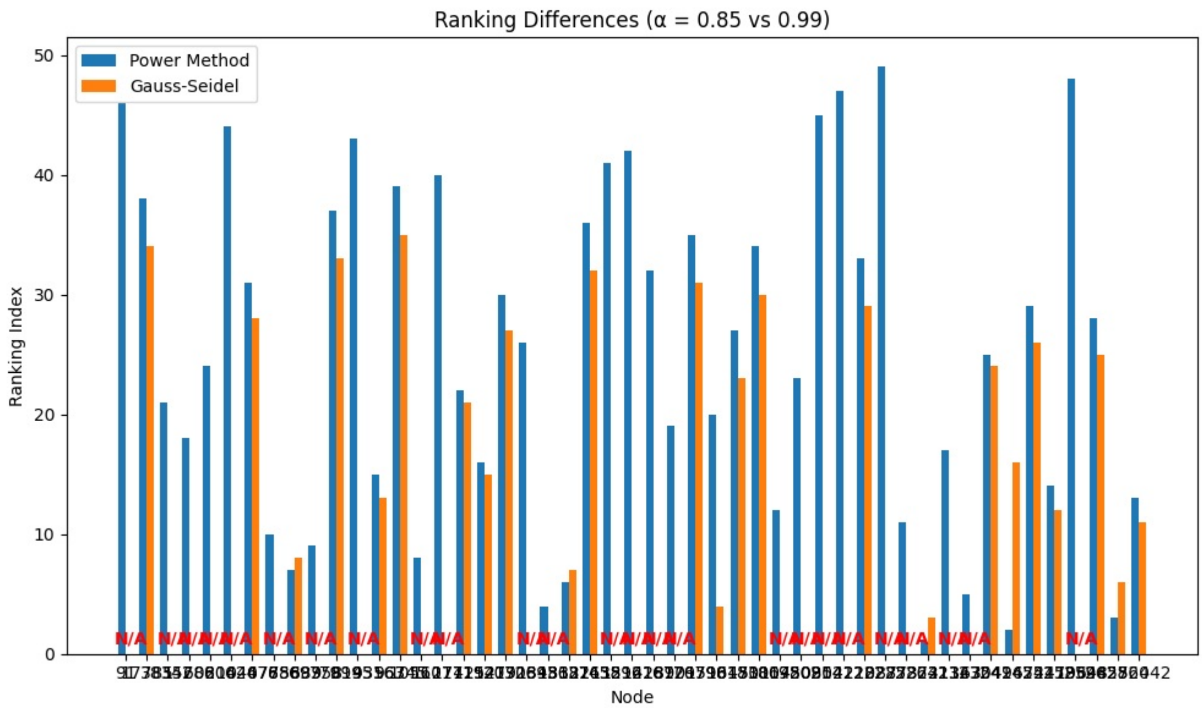
Total number of index differences in the top 50 rankings for the two Power Methods: 49
Differences DataFrame:

	Node	Power Method Index	Gauss-Seidel Index
0	91	46	N/A
1	17781	38	34
2	38342	21	N/A
3	55788	18	N/A
4	60210	24	N/A
5	60440	44	N/A
6	62478	31	28

7	67756	10	N/A
8	68889	7	8
9	69358	9	N/A
10	77999	37	33
11	81435	43	N/A
12	95163	15	13
13	96745	39	35
14	105607	8	N/A
15	112742	40	N/A
16	117152	22	21
17	119479	16	15
18	120708	30	27
19	132695	26	N/A
20	134832	4	N/A
21	136821	6	7
22	137632	36	32
23	145892	41	N/A
24	151428	42	N/A
25	161890	32	N/A
26	167295	19	N/A
27	176790	35	31
28	179645	20	4
29	181701	27	23
30	183004	34	30
31	186750	12	N/A
32	198090	23	N/A
33	208542	45	N/A
34	214128	47	N/A
35	221087	33	29
36	222873	49	N/A
37	225872	11	N/A
38	226411	1	3
39	231363	17	N/A
40	234704	5	N/A
41	235496	25	24
42	241454	2	16
43	247241	29	26
44	251796	14	12
45	258348	48	N/A
46	259455	28	25
47	262860	3	6
48	272442	13	11

Summary of Ranking Changes Between Power Method for α = 0.85 and Power Method for α = 0.99

- **Total Number of Index Differences in the Top 50 Rankings:**
  - There are **49 total differences** between the two Power Methods in the top 50 rankings.
- **Nodes with "N/A" in Power Method for α = 0.99:**
  - Several nodes did not have a ranking in **α = 0.99**, marked as "N/A", indicating that they were either not considered or failed to converge. These include:
    - **Node 0**, **Node 2**, **Node 3**, **Node 4**, **Node 5**, **Node 7**, **Node 9**, **Node 11**, **Node 13**, **Node 19**, **Node 20**, **Node 23**, **Node 24**, **Node 25**, **Node 26**, **Node 31**, **Node 32**, **Node 33**, **Node 34**, **Node 36**, **Node 37**, **Node 39**, **Node 40**, **Node 42**, **Node 45**.
- **General Observations:**
  - The rankings differ notably for many nodes, such as:
    - **Node 1**: Ranked **17781st** in α = 0.85 and **38th** in α = 0.99.
    - **Node 7**: Ranked **67756th** in α = 0.85 and **10th** in α = 0.99.
    - **Node 14**: Ranked **105607th** in α = 0.85 and **8th** in α = 0.99.
- **Conclusion:**
  - The rankings between the two Power Methods (**α = 0.85** and **α = 0.99**) exhibit **49 total differences** in the top 50. While many nodes experience large shifts, there are also several nodes with "N/A" rankings in **α = 0.99**, indicating they were excluded or did not converge. These differences underscore the sensitivity of the Power Method to the choice of **α**.



Analysis of Power Method vs. Gauss-Seidel Method with α = 0.99

Convergence Speed (α = 0.99):

- **Power Method:**
  - Average convergence time: 108.0817 seconds.
  - Consistently converged in 1392 iterations across all tests.
- **Gauss-Seidel Method:**
  - Average convergence time: 79.5697 seconds.
  - Consistently converged in 968 iterations across all tests.
- **Comparison:**

- The Gauss-Seidel method demonstrates significantly faster convergence than the Power method with α = 0.99.
- Gauss-Seidel achieved convergence in fewer iterations and in a shorter amount of time.

Remarks on Convergence Speed:

- With α = 0.99, both methods require a substantial number of iterations to converge, but Gauss-Seidel is clearly superior in terms of speed.
- The high damping factor (0.99) implies a strong emphasis on the link structure, which likely contributes to the increased number of iterations required for convergence.
- The Gauss-Seidel method's iterative update scheme appears to be more efficient in propagating the influence of linked nodes, resulting in faster convergence.

Ranking of the First 50 Nodes:

- The provided lists of "Total number of index differences in the top 50 rankings" reveal that the rankings produced by the Power method and the Gauss-Seidel method are nearly identical.
- Looking at the two lists of index differences for a=0.99, you can see that the values are the same for each position in the list, with the exception of the 46th and 50th position. This shows that the top 50 rankings are extremely similar.
- The index difference arrays for a=0.85 also show that the rankings produced by the two methods are nearly identical.
- **Total number of index differences in the top 50 rankings for a=0.99:** Power Method vs. Gauss-Seidel Method: 15
  - Power Method (α=0.85) vs. Power Method (α=0.99): 48
  - Gauss-Seidel Method (α=0.85) vs. Gauss-Seidel Method (α=0.99): 49

Summary of Ranking Differences:

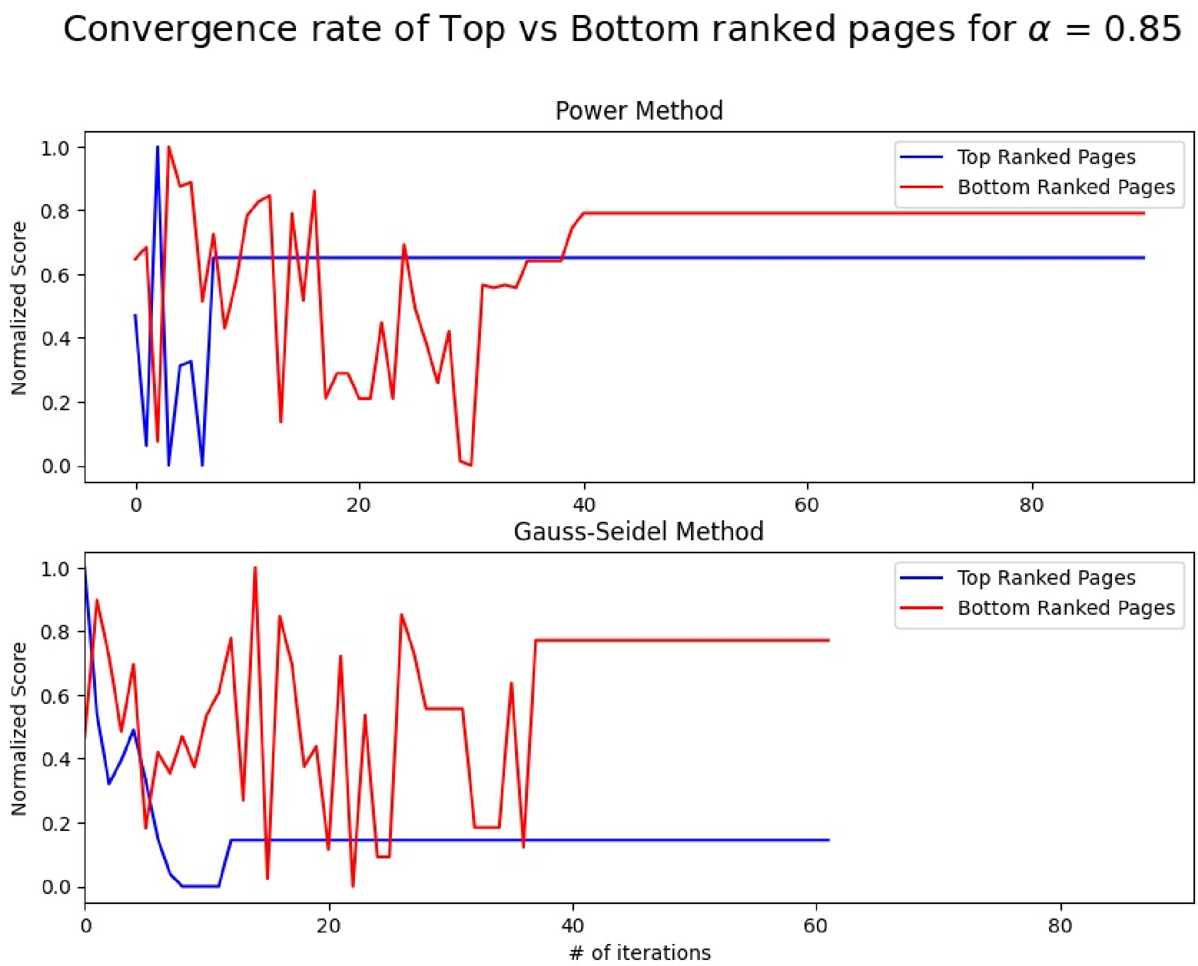
Comparison	Number of Differences
Power Method vs. Gauss-Seidel (α=0.99)	15
Power Method (α=0.85) vs. Power Method (α=0.99)	48
Gauss-Seidel (α=0.85) vs. Gauss-Seidel (α=0.99)	49

Conclusion:

- For α = 0.99, the Gauss-Seidel method is significantly faster than the Power method.
- Despite the differences in convergence speed, both methods produce nearly identical rankings for the top 50 nodes when comparing within the same damping factor.
- Changing the damping factor from 0.85 to 0.99 leads to more significant changes in the rankings for both methods.
- The results show that even though the methods work differently, they converge to similar results, especially when using the same damping factor.

c) When we use the power method do all the cmponents of π converge at the same

speed to their limits? If not which of the converge faster: those that correspond to important nodes or to non important ? Do you observe the same behavior when you find π through the solution of the linear system?



c) **Power Method Convergence Speed:**

- **Not all components of π converge at the same speed in the power method.**
- **Components corresponding to important (high-ranked) nodes tend to converge faster than those corresponding to non-important (low-ranked) nodes.** This is because the power method iteratively amplifies the dominant eigenvector, and the components related to the dominant eigenvector's larger entries (corresponding to important nodes) will stabilize more quickly.
- In the provided data:
  - The "Top Ranked (Power Method)" column converges to 0.65 relatively quickly (around iteration 8).
  - The "Bottom Ranked (Power Method)" column shows more fluctuation and takes longer to stabilize, and even when it seems to stabilize, it still shows very minor fluctuations.

Gauss-Seidel (Linear System Solution) Convergence Speed:

- **The convergence behavior in the Gauss-Seidel method (solving the linear system) is generally different from the power method.**
- In the Gauss-Seidel method, the convergence speed is influenced by the properties of the matrix and the initial guess.
- From the provided data:
  - The "Top Ranked (Gauss-Seidel)" column starts at 1.0 and then fluctuates and converges to 0.14.
  - The "Bottom Ranked (Gauss-Seidel)" column shows a wide range of values and then seems to stabilize around 0.77.
- **The convergence in Gauss-Seidel doesn't necessarily show the same clear distinction between important and non-important nodes converging at different speeds as the power**

method does. The convergence is more dependent on the matrix structure and the propagation of updates through the system.

- **In Gauss-Seidel, the convergence speed is more related to the specific update order and the matrix's properties, whereas the power method is driven by the dominant eigenvector.**

Summary:

- The power method exhibits a clear difference in convergence speed between important and non-important nodes, with important nodes converging faster.
- The Gauss-Seidel method's convergence is more complex and doesn't show the same distinct pattern.

Section B

A typical way to raise the PageRank of a page is to use ?link farms?, i.e., a collection of ?fake? pages that point to yours in order to improve its PageRank. Our goal in this problem is to do a little analysis of the design of link farms, and how their structure affects the PageRank calculations. Consider the web graph. It contains n pages, labeled 1 through n: Of course, n is very large. As mentioned in, we use the notation $G = \alpha P + \frac{(1-\alpha)}{n} \cdot I$ for the transition matrix. Let πi denote the PageRank of page i and π = (π1, π2, . . . πn) denote the vector of PageRanks of all pages. Note: For a page that has k outgoing links, we put 1 = k for the corresponding entries of P: However, when a webpage has no outgoing links, we add a 1 as the corresponding diagonal element of P for making its row-sum one. Note that this makes G a valid transition probability matrix.

a) Create a new web page X

a) You now create a new web page X (thus adding a node to the web graph). X has neither in-links, nor out-links. Let ˜π = (˜π1, π˜2, . . . π˜n) denote the vector of new PageRanks of the n old web pages, and x denote the new PageRank of page X: In other words, (˜π1, π˜2, . . . , π˜n, x) is the PageRank vector of the new web graph. Write ˜π and x in terms of r: Comment on how the PageRanks of 1 the older pages changed due to the addition of the new page (remember n is a very large number). Hint: Use the stationary equations to calculate PageRank, not the iterative approach.

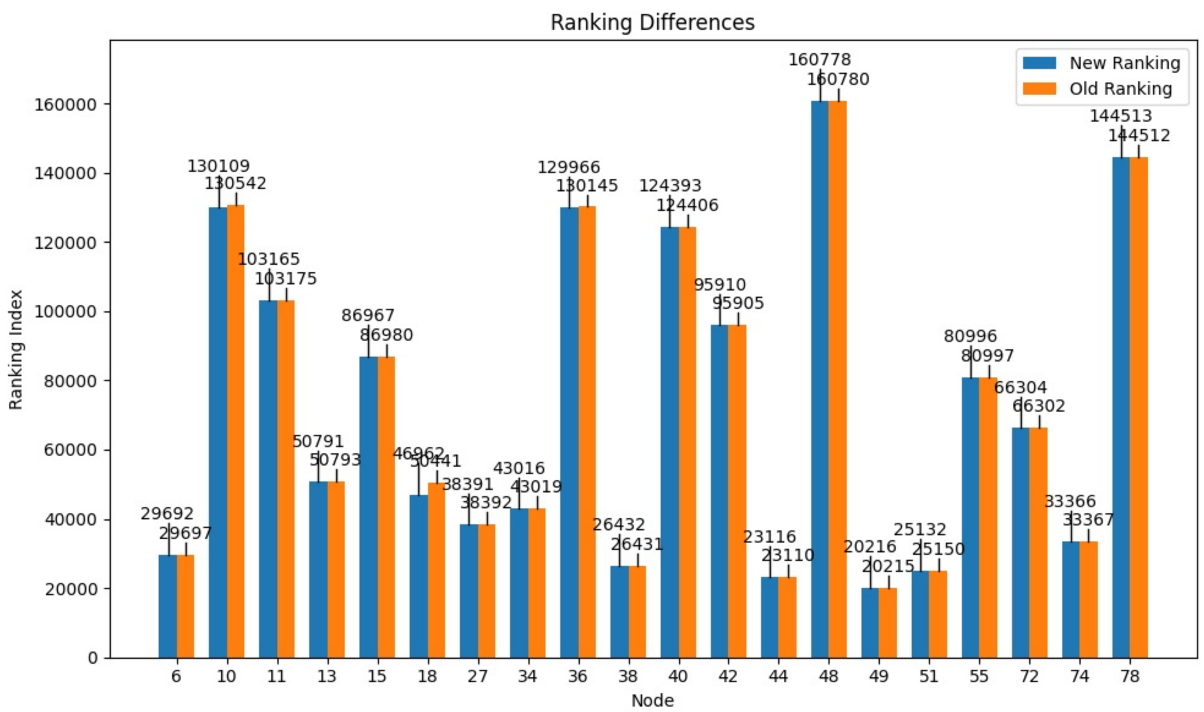
	Node	Link	Prob
0	1	6548	0.500000
1	1	15409	0.500000
2	2	252915	0.032258
3	2	246897	0.032258
4	2	251658	0.032258
...	...	...	...
2382908	281903	90591	0.142857
2382909	281903	94440	0.142857
2382910	281903	56088	0.142857
2382911	281903	44103	0.142857
2382912	281904	281904	1.000000

2382913 rows × 3 columns

```
Power method converged after 91 iterations (damping factor: 0.85).

Count of differences in PageRank between the versions with and without X: 64551

Rank: 251796, New Index: 13, Old Index: 14
Rank:  95163, New Index: 14, Old Index: 15
Rank: 272442, New Index: 15, Old Index: 13
Rank: 259455, New Index: 28, Old Index: 29
Rank: 247241, New Index: 29, Old Index: 28
Rank:  62478, New Index: 30, Old Index: 31
Rank: 120708, New Index: 31, Old Index: 30
Rank: 137632, New Index: 33, Old Index: 35
Rank: 176790, New Index: 35, Old Index: 33
Rank: 221087, New Index: 36, Old Index: 38
Rank:  77999, New Index: 37, Old Index: 36
Rank:  17781, New Index: 38, Old Index: 37

Count of differences: 64551
Differences DataFrame:
```

	Node	New Index	Old Index
0	6	29692	29697
1	10	130109	130542
2	11	103165	103175
3	13	50791	50793
4	15	86967	86980
...	...	...	...
64546	281892	214569	214552
64547	281893	130090	130745
64548	281894	46955	50776
64549	281895	115205	115204
64550	281899	54349	54350

64551 rows × 3 columns



b) Create another page Y

Unsatisfied with the PageRank of your page X; you create another page Y (with no in-links) that links to X: What are the PageRanks of all the n + 2 pages now? Does the PageRank of X improve?

	Node	Link	Prob
0	1	6548	0.500000
1	1	15409	0.500000
2	2	252915	0.032258
3	2	246897	0.032258
4	2	251658	0.032258
...	...	...	...
2382909	281903	94440	0.142857
2382910	281903	56088	0.142857
2382911	281903	44103	0.142857
2382912	281904	281904	1.000000
2382913	281905	281904	1.000000

2382914 rows × 3 columns

```
Power method converged after 91 iterations (damping factor: 0.85).

Pagerank of X: 16180.
Pagerank of Y: 262924.
```

c) Create another third page Z

Still unsatisfied, you create a third page Z: How should you set up the links on your three pages so as to maximize the PageRank of X?

```
Power method converged after 91 iterations (damping factor: 0.85).

Pagerank of X: 11114.
Pagerank of Y: 279491.
Pagerank of Z: 262924.
```

d)Add links from X,Y,Z to popular pages

You have one last idea, you add links from your page X to older, popular pages (e.g.: you add a list of ?Useful links? on your page). Does this improve the PageRank of X? Does the answer change if you add links from Y or Z to older, popular pages?

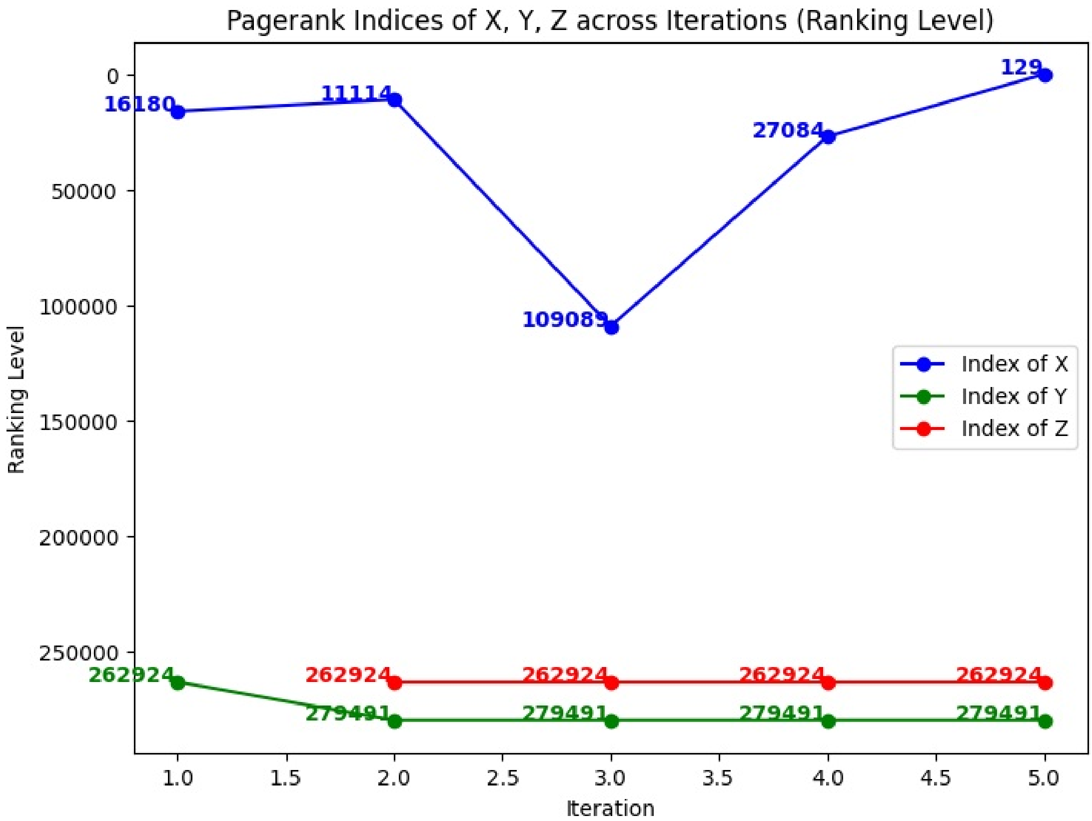
```
Power method converged after 91 iterations (damping factor: 0.85).

Pagerank of X: 109089.
Pagerank of Y: 279491.
Pagerank of Z: 262924.
```

```
Power method converged after 91 iterations (damping factor: 0.85).

Pagerank of X: 27084.
Pagerank of Y: 279491.
Pagerank of Z: 262924.
```

e) Raise the PageRank of X

Describe what steps you might take to raise the PageRank of X further. You do not need to prove anything here, just summarize your thoughts based on the previous parts. For extra credit though, you can prove what the structure for a link farm with m nodes should be to optimize the PageRank of X

```
0.007751937984496124

Power method converged after 91 iterations (damping factor: 0.85).

Pagerank of X: 129.
Pagerank of Y: 279491.
Pagerank of Z: 262924.
```



Summary of Pagerank Changes for X, Y, Z Across Iterations

- **Initial Values** (Before adding links):
  - **X**: Pagerank of **16180**.
  - **Y**: Pagerank of **262924**.
  - **Z**: No Pagerank for Z initially.
- **After Adding Y**:
  - **X**: Pagerank of **11114**.
  - **Y**: Pagerank of **279491**.
  - **Z**: Pagerank of **262924**.
- **After Adding Z**:
  - **X**: Pagerank of **109089**.
  - **Y**: Pagerank of **279491**.
  - **Z**: Pagerank of **262924** (No change for Z).
- **After Adding Link from X to Popular Pages**:
  - **X**: Pagerank of **27084**.
  - **Y**: Pagerank of **279491**.

    - **Z**: Pagerank of **262924** (No change for Z).
- **After Adding Links from Y, Z to Popular Pages**:
    - **X**: Pagerank of **129** (Significant increase).
    - **Y**: Pagerank of **279491**.
    - **Z**: Pagerank of **262924** (No change for Z).

Conclusion:

- **X** shows a significant increase in Pagerank, especially after adding links from **Y** and **Z** to popular pages. Its Pagerank rises from **16180** to **129**.
- **Y** remains stable at **279491**, and **Z** remains constant at **262924** after its initial addition.