Data Structures and Algorithms Spring 2024 — Problem Sets

by Maksim Al Dandan

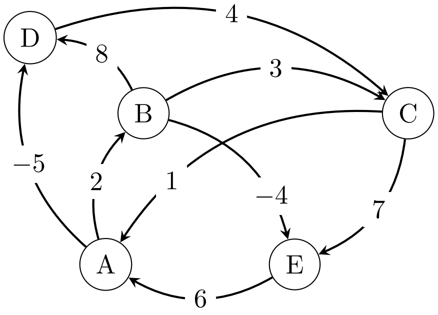
April 21, 2024

Week 13. Problem set

1 Task 1

1.1 Statement

Run the Floyd-Warshall algorithm [Cormen, §23.2] on the following graph. Use the alphabetic order of vertices. Show the state of distance matrix D after each iteration of outer loop in the algorithm. Since the graph has 5 vertices, you must provide five 5×5 matrices in your answer. No justification is required.



1.2 Solution

Initial matrix

	A	**B**	**C**	**D**	**E**
A	0	2	∞	-5	∞
B	∞	0	3	8	-4
C	1	∞	0	∞	7
D	∞	∞	4	0	∞
E	6	∞	∞	∞	0

Matrix after 1-st iteration

	A	**B**	**C**	**D**	**E**
A	0	2	∞	-5	∞
B	∞	0	3	8	-4
C	1	3	0	-4	7
D	∞	∞	4	0	∞
E	6	8	∞	1	0

Matrix after 2-nd iteration

	A	**B**	**C**	**D**	**E**
A	0	2	5	-5	-2
B	∞	0	3	8	-4
C	1	3	0	-4	-1
D	∞	∞	4	0	∞
E	6	8	11	1	0

Matrix after 3-rd iteration

	A	**B**	**C**	**D**	**E**
A	0	2	5	-5	-2
B	4	0	3	-1	-4
C	1	3	0	-4	-1
D	5	7	4	0	3
E	6	8	11	1	0

Matrix after 4-th iteration

	A	**B**	**C**	**D**	**E**
A	0	2	-1	-5	-2
B	4	0	3	-1	-4
C	1	3	0	-4	-1
D	5	7	4	0	3
E	6	8	5	1	0

Matrix after 5-th iteration

	A	**B**	**C**	**D**	**E**
A	0	2	-1	-5	-2
B	2	0	1	-3	-4
C	1	3	0	-4	-1
D	5	7	4	0	3
E	6	8	5	1	0

2 Task 2

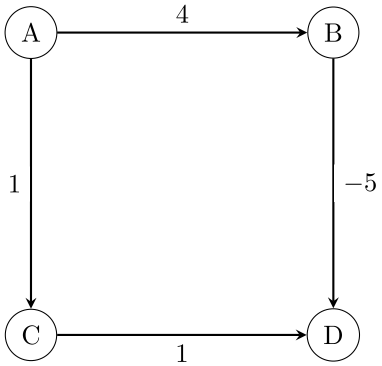
2.1 Statement

Provide a graph with exactly 4 vertices (A, B, C, D) and 4 weighted edges, such that Dijkstra's algorithm [Cormen, §22.3] does **not** give a correct shortest distance for at least one vertex:

1. Provide the graph (the graph must be rendered in a clear way, text representation is not enough); Weights must be (small) integers.
2. Provide the result of Dijkstra's algorithm for each vertex: any shortest path **and** corre- sponding total weight;
3. Provide the correct shortest path and corresponding total weight for each vertex;
4. Explain why Dijkstra's algorithm did not provide the correct answer (specifically for your example, generic justification is not accepted).

2.2 Solution

Graph with 4 vertices and 4 weighted edges



Result of Dijkstra's algorithm for each vertex

- From A to B: Path A-B with total weight 4.
- From A to C: Path A-C with total weight 1.
- From A to D: Path A-B-D with total weight -1 (Incorrect due to negative weight).

Correct shortest path and corresponding total weight for each vertex

- From A to B: Path A-B with total weight 4.
- From A to C: Path A-C with total weight 1.
- From A to D: Path A-C-D with total weight 5.

Explanation

Dijkstra's algorithm did not provide the correct answer for the vertex D because it cannot handle graphs with negative weight edges properly. In this example, the algorithm prematurely concludes that the shortest path to D is through B because it does not revisit vertices with updated distances once they have been marked as visited. This leads to the incorrect assumption that the path A-B-D is shorter than it actually is, ignoring the fact that the path A-C-D is the correct shortest path when considering all edges, including those with negative weights.

3 Task 3

3.1 Statement

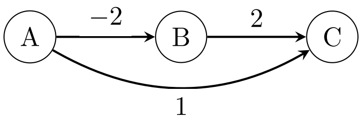
Since Dijkstra's shortest paths algorithm [Cormen, §22.3] does not work with negative edges in general, consider an algorithm that, for a given graph G, if it has a negative edge, finds the minimum edge in G with the weight $(-W)$ and adds $(+W)$ to all edges in the original graph, resulting in a new graph G^{+W}. Then the modified algorithm runs Dijkstra's algorithm on G^{+W}. Are the resulting shortest paths in G^{+W} also shortest in G? If yes, prove it. If no, provide a concrete counterexample and a justification.

3.1.1 Solution

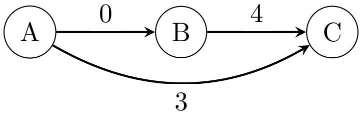
Answer: No, the resulting shortest paths in G^{+W} are not necessarily the shortest paths in the original graph G.

Counterexample:

Consider a graph G with three vertices A, B, and C, and three edges: $A \to B$ with weight -2, $B \to C$ with weight 2, and $A \to C$ with weight 1.



In G, the shortest path from A to C is directly from A to C with a total weight of 1.
Now, we find the minimum edge weight, which is -2, and add 2 to all edges in G to get G^{+2}:



In G^{+2}, the shortest path from A to C is now $A \to B \to C$ with a total weight of 4. However, when we subtract the added weight $(2 \times 2 = 4)$ from this total, we get 0, which is not the shortest path weight in the original graph G.

Justification:

The process of adding a constant weight to all edges in a graph with negative edges can alter the relative weights of different paths. While this transformation makes all edges non-negative, allowing Dijkstra's algorithm to be applied, it does not preserve the original shortest path properties of the graph. This is because the added weight disproportionately affects shorter paths less than longer paths, potentially making a longer path seem shorter after the transformation. Therefore, the resulting shortest paths in G^{+W} are not guaranteed to be the shortest paths in the original graph G.

References

[Cormen] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein. *Introduction to Algorithms, Fourth Edition.* The MIT Press 2022