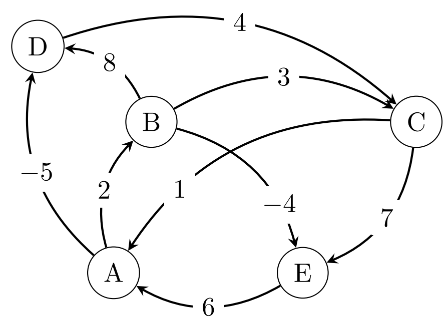
Data Structures and Algorithms Spring 2024 — Problem Sets

by Kiril Efimovichh

June 24, 2025

Week 13. Problem set

1. 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.



Answer:

Dist A:	A	B	C	D	E
A	0	2	+INF	-5	+INF
B	+INF	0	3	8	-4
C	1	3	0	-4	7
D	+INF	+INF	4	0	+INF
E	6	8	+INF	1	0

Dist B:	A	B	C	D	E
A	0	2	5	-5	-2
B	+INF	0	3	8	-4
C	1	3	0	-4	-1
D	+INF	+INF	4	0	+INF
E	6	8	11	1	0

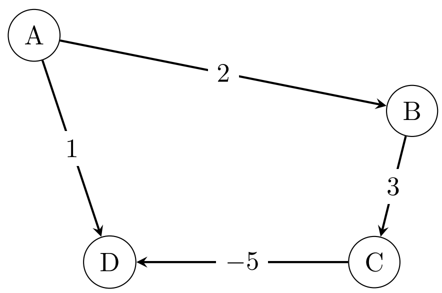
Dist C:	A	B	C	D	E
A	0	2	5	-5	-2
B	4	0	3	-1	-4
C	1	3	0	-4	-1
D	5	7	4	0	3
E	6	8	11	1	0

Dist D:	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

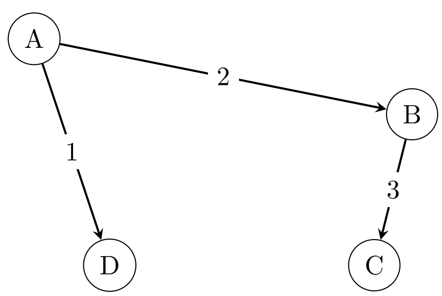
Dist E:	A	B	C	D	E
A	0	2	-1	-5	-2
B	2	0	3	-3	-4
C	1	3	0	-4	-1
D	5	7	4	0	3
E	6	8	5	1	0

2. 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:

   (a) Provide the graph (the graph must be rendered in a clear way, text representation is not enough); Weights must be (small) integers.
   Answer:

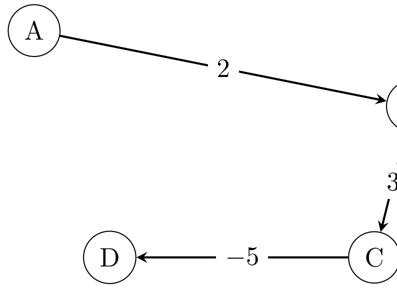


   (b) Provide the result of Dijkstra's algorithm for each vertex: any shortest path **and** corresponding total weight;
   Answer:



   Djikstra:
   A → A = 0;
   A → B = 2;
   A → C = 5;
   A → D = 1;

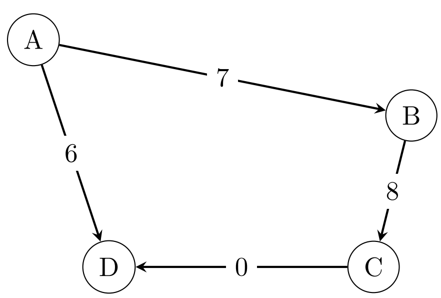
   (c) Provide the correct shortest path and corresponding total weight for each vertex;
   Answer:



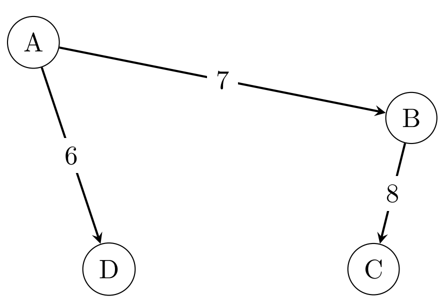
Shortest paths:
A → A = 0;
A → B = 2;
A → C = 5;
A → D = 0;

(d) Explain why Dijkstra's algorithm did not provide the correct answer (specifically for your example, generic justification is not accepted).
Answer:
Dijkstra's algorithm doesn't work with edges with negatives weights. This is because Dijkstra's algorithm works by property of triangle inequality: $\delta(S,v) \leq \delta(S,u) + \omega(u,v)$. In this example in doesn't work for S be equal to vertex A and v be equal to vertex D:
$\delta(A,D) \leq \delta(A,BC) + \omega(BC,D)$
$1 \leq 2 - 2$
$1 \leq 0$ - FALSE

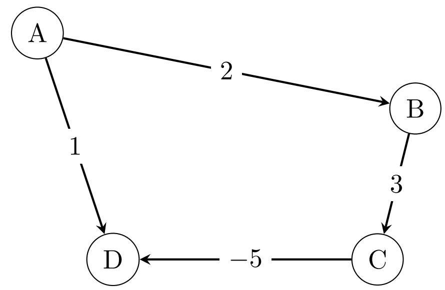
3. Since Dijkstra's shortest paths algorithm [Cormen, §22.3] does not work with negative edges in general, consider an algorithm that, for a given graph G, if it has a negative edge, finds the minimum edge in G with the weight $(-W)$ and adds $(+W)$ to all edges in the original graph, resulting in a 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.
Answer:
G^{+w}:



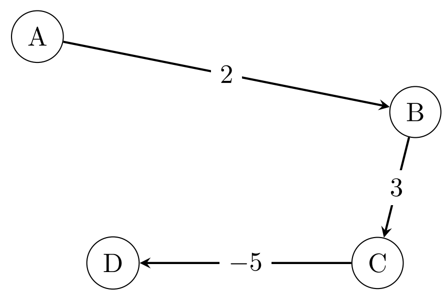
Result:



We can find counterexample:
G:



Result:



References

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