

Quiz 6 - Dijkstra's

Due Date February 11
Name **Your Name**
Student ID **Your Student ID**

Contents

1	Instructions	1
2	Honor Code (Make Sure to Virtually Sign)	2
3	Standard 6 - Dijkstra's	3
3.1	Problem 2	3

1 Instructions

- The solutions **should be typed**, using proper mathematical notation. We cannot accept hand-written solutions. Here's a short intro to \LaTeX .
- You should submit your work through the **class Canvas page** only. Please submit one PDF file, compiled using this \LaTeX template.
- You may not need a full page for your solutions; pagebreaks are there to help Gradescope automatically find where each problem is. Even if you do not attempt every problem, please submit this document with no fewer pages than the blank template (or Gradescope has issues with it).
- You **may not collaborate with other students**. **Copying from any source is an Honor Code violation. Furthermore, all submissions must be in your own words and reflect your understanding of the material.** If there is any confusion about this policy, it is your responsibility to clarify before the due date.
- Posting to **any** service including, but not limited to Chegg, Discord, Reddit, StackExchange, etc., for help on an assignment is a violation of the Honor Code.
- You **must** virtually sign the Honor Code (see Section 2). Failure to do so will result in your assignment not being graded.

2 Honor Code (Make Sure to Virtually Sign)

Problem 1.

- My submission is in my own words and reflects my understanding of the material.
- I have not collaborated with any other person.
- I have not posted to external services including, but not limited to Chegg, Discord, Reddit, StackExchange, etc.
- I have neither copied nor provided others solutions they can copy.

Agreed (signature here). I agree to the above. Ethan Richman.

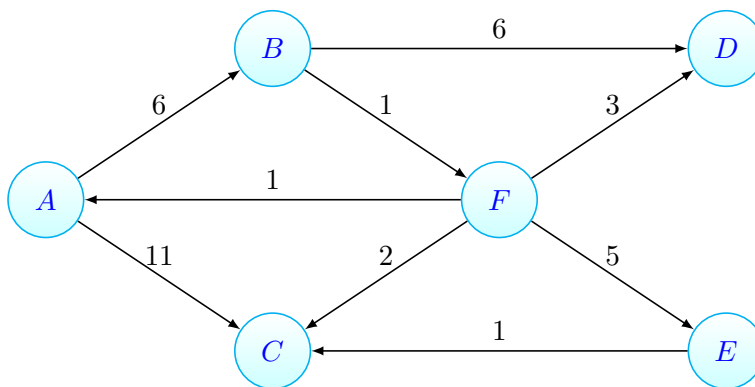
□

3 Standard 6 - Dijkstra's

3.1 Problem 2

Problem 2. Consider the weighted graph $G(V, E, w)$ pictured below. Work through Dijkstra's algorithm on the following graph, using the source vertex A .

- Clearly include the contents of the priority queue, as well as the distance from A to each vertex at each iteration.
- If you use a table to store the distances, clearly label the keys according to the vertex names rather than numeric indices (i.e., `dist['B']` is more descriptive than `dist['1']`).
- You do **not** need to draw the graph at each iteration, though you are welcome to do so. [This may be helpful scratch work, which you do not need to include.]



Answer. We proceed as follows:

a: Our source node is A . We begin by setting the found distances for each node other than A to ∞ . We also set A as processed.

b: We begin by examining A 's neighbors, updating their found distances and placing into the priority queue so: $Q=[(\{A,B\},6), (\{A,C\},11)]$. And the found distances from A are currently ($B : 6, C : 11, D : \infty, E : \infty, F : \infty$).

c: We poll the first vertex from the priority queue which is B with found distance 6 and mark B as processed. Now we examine B 's unprocessed neighbors which is D and F and add them to the queue. $Q=[(\{B,F\},1), (\{B,D\},6), (\{A,C\},11)]$. And found the distances from A are currently ($B : 6, C : 11, D : 12, E : \infty, F : 7$)

d: We then poll the next vertex from the queue which is F and mark it as processed. Now we examine F 's unprocessed neighbors which include D, E, C and add them to the queue. $Q=[(\{F,C\},2), (\{F,D\},3), (\{F,E\},5), (\{B,D\},6), (\{A,C\},11)]$ and found distances from A are currently ($B:6, C:9, D:10, E:12, F:7$)

e: We then poll vertex C from the queue and mark it as processed and since it does not have any unprocessed neighbors so. $Q=[(\{F,D\},3), (\{F,E\},5), (\{B,D\},6), (\{A,C\},11)]$ and the distances from A are currently ($B:6, C:9, D:10, E:12, F:7$)

f: We then poll vertex D from the queue and mark it as processed and it does not have any unprocessed neighbors so $Q=[(\{F,E\},5), (\{B,D\},6), (\{A,C\},11)]$ and the distances from A are currently ($B:6, C:9, D:10, E:12, F:7$)

g: We then poll vertex E from the queue and mark it as processed and now every vertex is marked as processed to the algorithm terminates and the final distances from A is ($B:6, C:9, D:10, E:12, F:7$)

□