Quiz 8 – Kruskal's Algorithm

Due Date .... February 18
Name .... **Julia Troni**
Student ID **109280095**

Contents

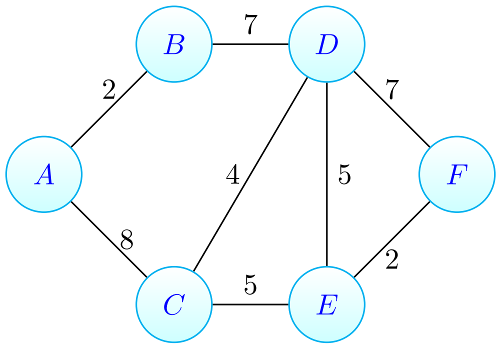
1 Instructions

- 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.

2 Standard 8- Kruskal's Algorithm

2.1 Problem 1

Problem 1. Consider the following graph $G(V, E, w)$. Clearly indicate the order in which Kruskal's algorithm adds the edges to the minimum-weight spanning tree. You may simply list the order of the edges; it is not necessary to exhibit the state of the algorithm (i.e., the disjoint-set data structure) at each iteration.



Answer. Kruskal's algorithm begins by initializing the intermediate spanning forest $\mathcal{F}$ to contain all the vertices but no edges. Then, it creates a priority queue in which it adds all of the edges of the graph, ordered from lowest weight to highest weight.
For this graph, the edges will be added to the queue such that initially the priority queue is
$Q = [(\{A, B\}, 2), (\{E, F\}, 2), (\{C, D\}, 4), (\{C, E\}, 5),$
$(\{D, E\}, 5), (\{B, D\}, 7), (\{D, F\}, 7), (\{A, C\}, 8)]$

Then the algorithm polls each edge from the priority queue and checks if it creates a cycle. If it does NOT create a cycle, the edge is added to $\mathcal{F}$. However, if it does create a cycle, meaning the two vertices are on the same component of $\mathcal{F}$, then the edge will NOT be added to $\mathcal{F}$.
The algorithm terminates when $\mathcal{F}$ spans all vertices, and at which point $\mathcal{F}$ has $|V(G)| - 1 = 6 - 1 = 5$ edges and is the MST of the graph.

For this graph, and using the ordering of the initial priority queue the MST $\mathcal{F}$ contains the following edges
$\mathcal{F} = [(\{A, B\}, 2), (\{E, F\}, 2), (\{C, D\}, 4), (\{C, E\}, 5), (\{B, D\}, 7)]$
The edges are added in this order:

1. ({A,B}, 2)
2. ({E,F}, 2)
3. ({C,D}, 4)
4. ({C,E}, 5)
5. ({B,D}, 7)

□