CS 3510 – Assignment 5

Due Friday, July 22, 2022 at 11:59pm on Canvas

Instructor: Shahrokh Shahi

- Please type your answers (LaTeX is highly recommended) and upload a single PDF file named `<Your-GT-Account>.pdf`, e.g., GBurdell3.pdf, including all your answers. You can submit multiple times. Canvas keeps track of the submissions and append a version number when you re-submit. We always grade your most recent submissions.
- Please read the policies, and do not forget to acknowledge your collaborators and cite your references.
- If you do not understand a question, please ask on Piazza or come to office hours well ahead of the due date.
- It is recommended to start reviewing the course material by reading the lecture slides and reviewing the demo codes. Then, the suggested readings from textbooks and solving the practice problems can provide the additional preparation for solving the homework problems. Please note, for the textbook readings, you do not need to cover the topics which have not been covered in the lectures.
- If you need to draw a figure, you can draw it on paper, take a picture, and include it in your submission. Please make sure the image quality/resolution is good and the numbers are readable.

Suggested Reading

	CLRS	KT
Section(s)	Chapter 24, 25, 26	Chapter 4.4, 6.8, 7

Suggested Practice Problems

	CLRS	KT
Practice problems	Exercise: 24.1-(1,3), 24.3-(1,6), 24.5-(5), 25.2-(1,4,6), 26.2-(2,4) Problems: 24-1, 26-4	Solved Exercise: Exercise: 1-5

Additional reading and problems:

– DPV (Chapter 4, 6, 7.2)

1 Shortest Path Problem (18 pts)

Given an undirected graph $G = (V, E)$ with positive edge weights and two nodes $s, t \in V$, design an efficient algorithm to determine the set of all edges that lie on at least one shortest path from s to t.

(a) (8 pts) Explain your algorithm in words. No pseudocode is required.

(b) (6 pts) Justify correctness of your approach.

(c) (4 pts) Discuss the running time of your algorithm.

Solution:

(a) Design: We start by finding the distance from s to every other node, denoted by $d_s(a)$ for $a \in V$. We then find the distance from t to every other node, denoting those lengths as $d_t(a)$. Note that $d_s(t) = d_t(s)$. Consider some edge $e = (a, b) \in E$, where $a, b \in V$ are the vertices connected by e. The shortest path length from s to t that goes through e is the minimum of: $d_s(a) + w(e) + d_v(b)$ and $d_s(b) + w(e) + d_v(a)$.

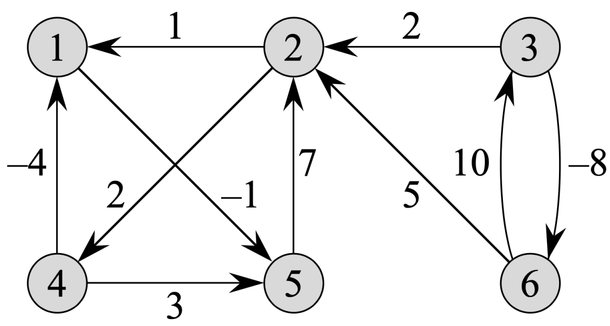
If this length is equal to $d_s(t)$, then e lies on a shortest path from s to t. We can repeat the same algorithm for every edge in the graph.

(b) Correctness: To see why the algorithm is correct, notice that the shortest path from s to t through an edge e must enter one of the two end vertices of e, traverse the edge, and move to t, the end point of the path. The minimum we took accounts for the two possibilities to do this. Clearly, an edge is on a shortest path if such path is of optimal (i.e.: minimum) length.

(c) Running time: We run Dijkstra's twice, which takes $O((E + V)\log(V)) \approx O(E\log(V))$ time. Note that since Dijkstra maintains a list of shortest distances between two nodes, we can use it compute the lengths in the manner specified above. That takes $O(E)$ time as we have to iterate through all edges. In total, the algorithm takes $O((E+V)\log(V)) \approx O(E\log(V))$ time as the second (linear) time is dominated by the runtime of Dijkstra's algorithm.

2 Shortest Path Problem (16 pts)

Run the `Floyd-Warshall` algorithm on the following weighted directed graph, and give the $n \times n$ matrix $D^{(k)}$ obtained at the end of iterations $k = 0, 1, 2, \text{and}, 3$, where $k = 0$ is the base case and $k = 1, 2, 3$ denotes the first three iterations.



Solution:

Note you were only asked for $k = 0, 1, 2, \text{and}, 3$.

k = 0 (base case)

0	∞	∞	∞	-1	∞
1	0	∞	2	∞	∞
∞	2	0	∞	∞	-8
-4	∞	∞	0	3	∞
∞	7	∞	∞	0	∞
∞	5	10	∞	∞	0

k = 1

0	∞	∞	∞	-1	∞
1	0	∞	2	0	∞
∞	2	0	∞	∞	-8
-4	∞	∞	0	-5	∞
∞	7	∞	∞	0	∞
∞	5	10	∞	∞	0

k = 2

0	∞	∞	∞	-1	∞
1	0	∞	2	0	∞
3	2	0	4	2	-8
-4	∞	∞	0	-5	∞
8	7	∞	9	0	∞
6	5	10	7	5	0

k = 3

0	∞	∞	∞	-1	∞
1	0	∞	2	0	∞
3	2	0	4	2	-8
-4	∞	∞	0	-5	∞
8	7	∞	9	0	∞
6	5	10	7	5	0

k = 4

0	∞	∞	∞	-1	∞
-2	0	∞	2	-3	∞
0	2	0	4	-1	-8
-4	∞	∞	0	-5	∞
5	7	∞	9	0	∞
3	5	10	7	2	0

k = 5

0	6	∞	8	-1	∞
-2	0	∞	2	-3	∞
0	2	0	4	-1	-8
-4	2	∞	0	-5	∞
5	7	∞	9	0	∞
3	5	10	7	2	0

k = 6

0	6	∞	8	-1	∞
-2	0	∞	2	-3	∞
-5	-3	0	-1	-6	-8
-4	2	∞	0	-5	∞
5	7	∞	9	0	∞
3	5	10	7	2	0

3 Flow Network (16 pts)

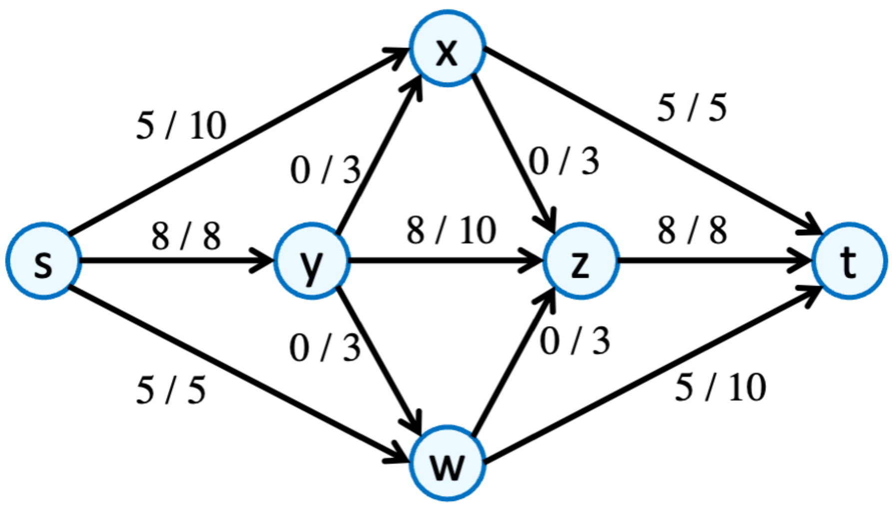
The following figure shows a flow network on which an $s-t$ flow has been computed. The capacity $c(e)$ and the amount of flow sent on each edge $f(e)$ appear as labels next to the edge.

(a) (2 pts) What is the value of this flow?

(b) (2 pts) Find a minimum $s-t$ cut in this flow network, and also say what its capacity is.

(c) (2 pts) Is the given flow a maximum (s,t) flow in this graph? Explain why?

(d) (10 pts) Assume the given flow is obtained at the middle of running the Ford-Fulkerson algorithm. Give the residual graph at this point. Then, augment flow along path $P : s \to x \to z \to y \to w \to t$, and give the value of the obtained flow.



Solution:

(a) The value of the flow is 5+8+5 = 18

(b) Consider $A = s, x$. Then the cut $(A, V - A)$ is a minimum cut and has capacity of 21.

(c) No, the value if the flow is less than min cut capacity.

(d) See the following figures:

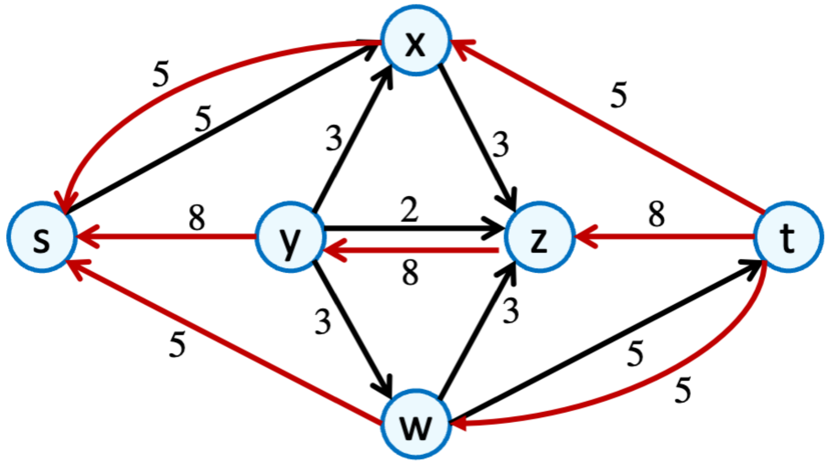


Figure 1: The residual graph

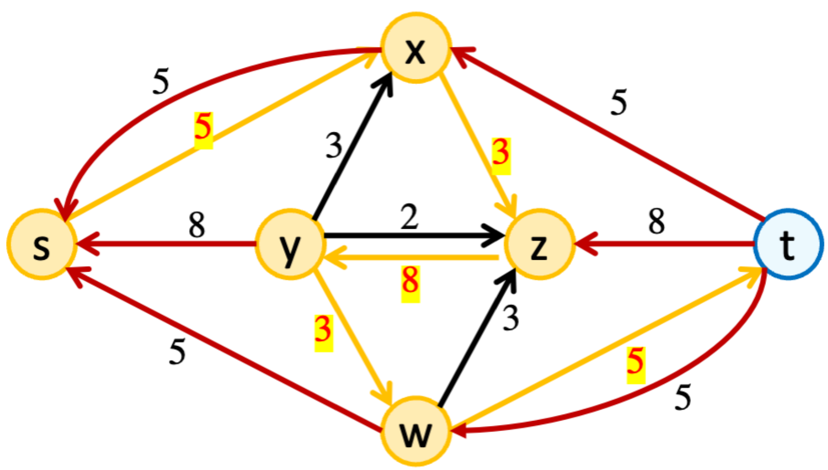


Figure 2: The augmenting path

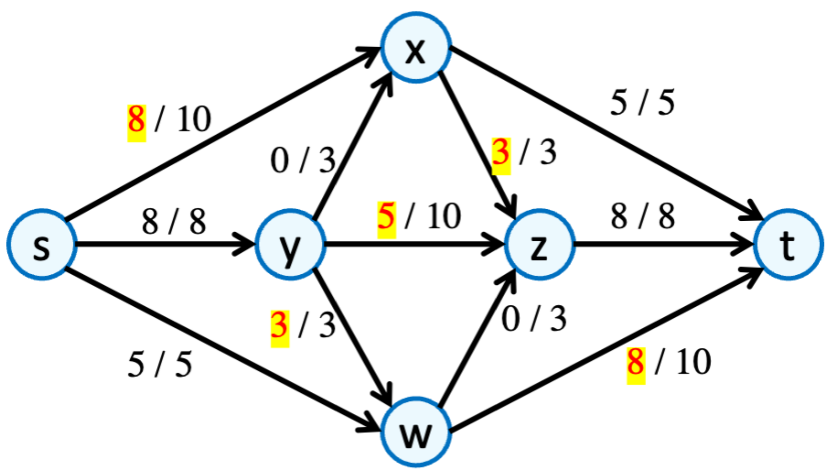


Figure 3: The obtained flow. Flow value = 5+8+8=21