Algorithms: Assignment #10

Tanvi Magdum

magdum.t@northeastern.edu

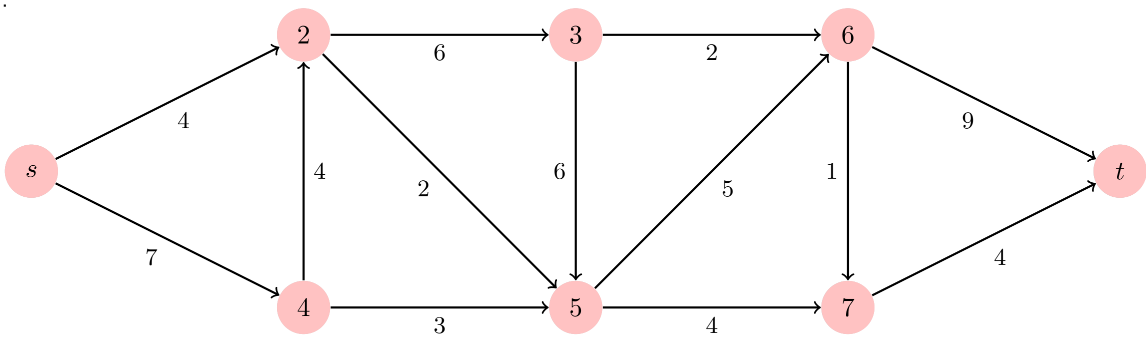
March 27, 2023

Answers

1 Problem 1 Solution :

Based upon given information, following is the flow graph of the new castle -

.

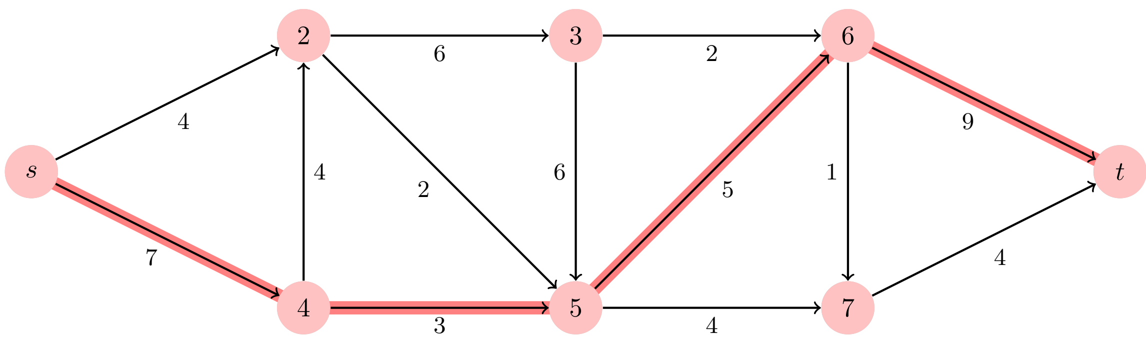


1.1 Problem 1.1 Solution :

The shortest augmenting path is the path with the fewest number of edges in the residual graph from the source node s to the sink node t. An augmenting path is a path from s to t in the residual graph along which we can increase the flow. The bottleneck of an augmenting path is the minimum capacity of all the edges along that path. Using greedy approach, we can choose the edge with the highest capacity at each step, while also ensuring that the path remains the shortest possible. It is important to consider all possible paths and choose the one with the shortest length and highest bottleneck capacity. Starting from the source node s, we choose the edge with the highest capacity, which takes us to a neighboring node. At each subsequent node, we again choose the edge with the highest capacity that leads to the shortest possible path to the sink node t.

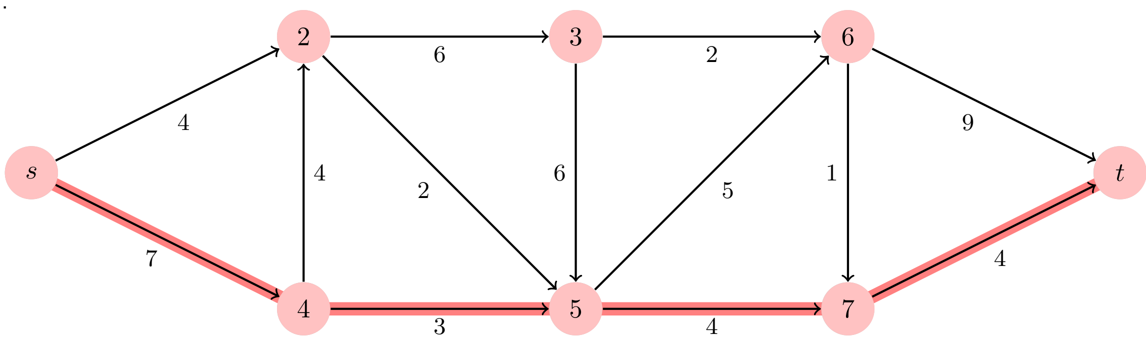
The path s -> 4 -> 5 -> 6 -> t with a bottleneck capacity of 4. We start at the source node S and choose the edge with the highest capacity, which is the edge s-4 with capacity 7. We then move to node 4, where we have two options: the edge 4-5 with capacity 3 and the edge 4-2 with capacity 4. However, we must choose the edge that leads to the shortest path to the sink node t. Since the path s -> 4 -> 5 -> 7 -> t has a shorter length than the any path s -> 4 -> 2 -> ... -> t, we choose the edge 4-5 with capacity 3. From node 5, we choose the edge with the highest capacity, which is 5-6 with capacity 5. Finally, we reach the sink node t, edge 6-t, where the path completes.

.



Now, if we use a different approach, we might find a different path which would also be the shortest augmenting path. For instance, here s -> 4 -> 5 -> 7 -> t is also the shortest augmenting path. Since we used a particular approach, we found out a path which has higher capacity and shortest distance between source and sink. If we use algorithm like BDS to calculate the shortest path, then it would give a different answer. But anyhow, there would be a consistency between lengths of all the possible shortest paths. Here the we need 4 edges for the shortest augmenting path.

.



1.2 Problem 1.2 Solution :

The goal here is to find the highest capacity augmenting path and its bottleneck for the given graph. An augmenting path is a path from the source to the sink with residual capacity along all of its edges. The highest capacity augmenting path is the path from the source to the sink with the highest bottleneck capacity. In other words, it is the path that can allow the maximum possible increase in the flow from the source to the sink. So, to find the highest capacity augmenting path, we need to find a path from the source node S to the sink node T with the highest bottleneck capacity.

The path s -> 4 -> 5 -> 6 -> t has the highest capacity augmenting path among all possible augmenting paths from the source s to the sink t. The capacities of the pipes along this path are -

c(s-4) = 7

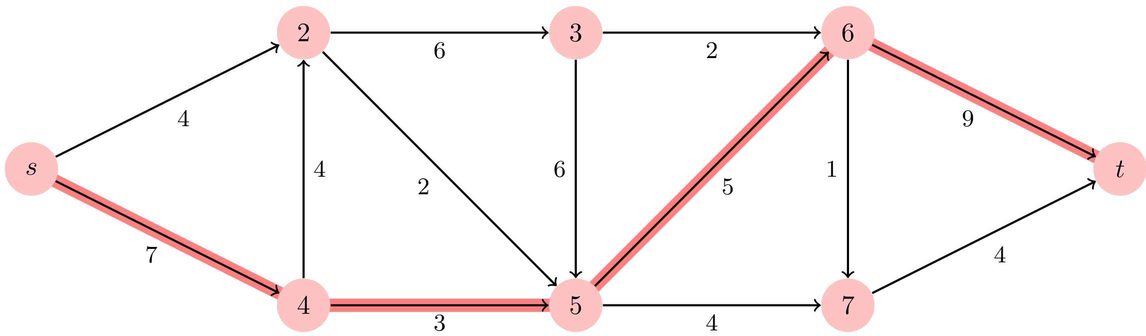
c(4-5) = 3

c(5-6) = 5

c(6-t) = 9

The capacity of a path is the minimum capacity of all the edges in the path. The bottleneck capacity of a path is the maximum capacity of all the edges in the path. The maximum amount of flow that can be sent along a path is limited by the bottleneck capacity of the path. Therefore, the bottleneck capacity of this path is 3 (the capacity of pipe 4->5). Increasing the flow along this path by 3 will result in the maximum possible increase in the overall flow. We can send at most 3 units of flow through this path, even if the other edges in the path have higher capacities. Also, the total capacity of this path is 24.

.



In this graph, there might be paths having greater bottleneck capacity, but they wouldn't accommodate more flow than the current chosen path, as the individual edge capacities are greater in this path. For example, s -> 2 -> 3 -> 5 -> 7 -> t is a path with bottleneck capacity 4, which is greater than the bottleneck capacity 3 of current path. But, the capacities of the edges in the path s -> 2 -> 3 -> 5 -> 7 -> t are smaller than the corresponding capacities in the path s -> 4 -> 5 -> 6 -> t. For instance, the path s -> 4 -> 5 -> 6 -> t has an edge with capacity 5 (the edge from 5 to 6), while the path s -> 2 -> 3 -> 5 -> 7 -> t does not have any edge with a capacity higher than 4. Therefore, even though the bottleneck capacity of s -> 2 -> 3 -> 5 -> 7 -> t is higher than that of s -> 4 -> 5 -> 6 -> t, the latter path can accommodate more flow because it has edges with higher capacities.

1.3 Problem 1.3 Solution :

To determine the maximum flow of the given flow graph using the Ford-Fulkerson algorithm, we need to repeatedly find the highest capacity augmenting path and increase the flow along this path until no more augmenting paths can be found. The algorithm is as below -

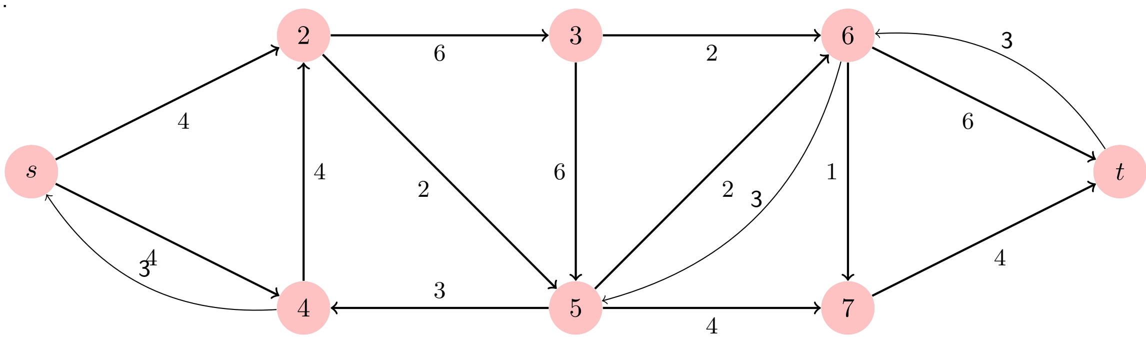
1. Initialize the flow on all edges to 0.
2. While there exists a path from s to t in the residual graph -
   (a) Find the highest capacity augmenting path from s to t.
   (b) Increase the flow along this path by the bottleneck capacity.
   (c) Update the residual graph by subtracting the flow from the capacities of the forward edges and adding the flow to the capacities of the backward edges.
3. The maximum flow is the sum of the flow leaving the source s.

Now, let G = {V, E}, where V = {s, 2, 3, 4, 5, 6, 7, t} and E = {s-2, s-4, 2-3, 2-5, 4-2, 4-5, 3-5, 3-6, 5-6, 5-7, 6-7, 6-t, 7-t}

Capacities : c(s-2) = 4, c(s-4) = 7, c(2-3) = 6, c(2-5) = 2, c(4-2) = 4, c(4-5) = 3, c(3-5) = 6, c(3-6) = 2, c(5-6) = 5, c(5-7) = 4, c(6-7) = 1, c(6-t) = 9, c(7-t) = 4

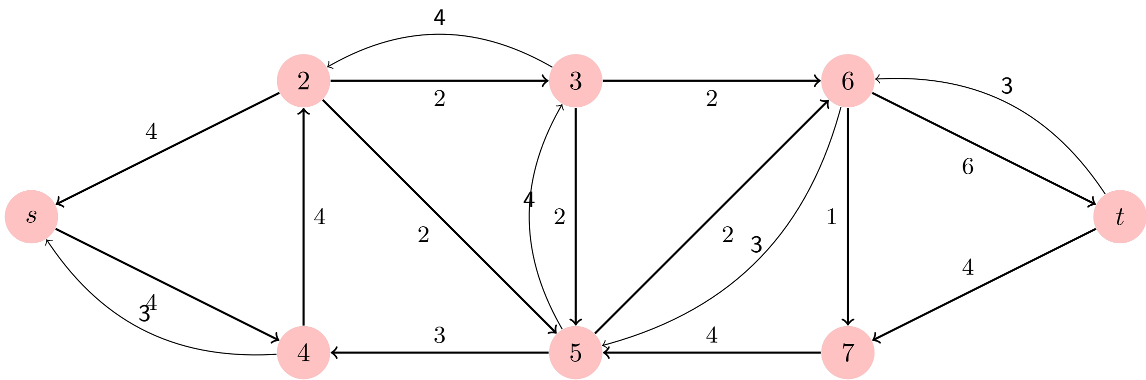
Now let us choose the highest capacity augmenting path from s to t. It is s -> 4 -> 5 -> 6 -> t. We need to send 3 units from this path, because edge 4-5 cannot accomodate more flow than 3 units. So, the residual graph will look like -

.



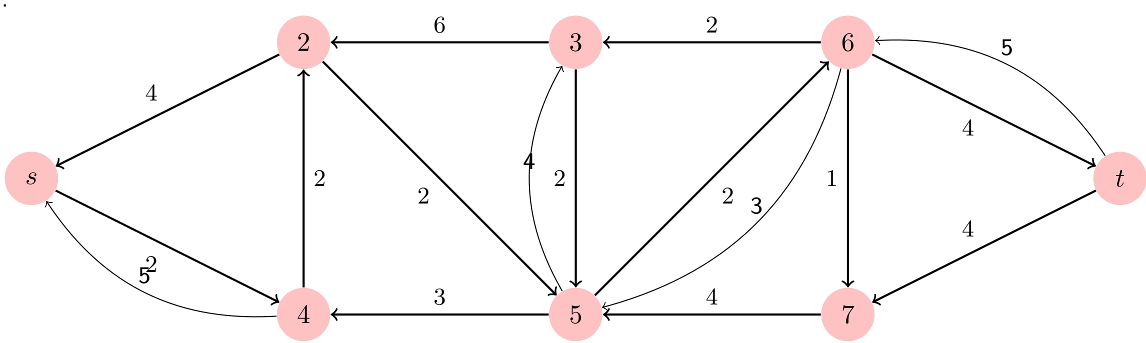
Flow value = 3

Let us chose the next path. Consider path s -> 2 -> 3 -> 5 -> 7 -> t. We can send 2 units through this path as edge s-2 has max capacity of 4. So the graph -



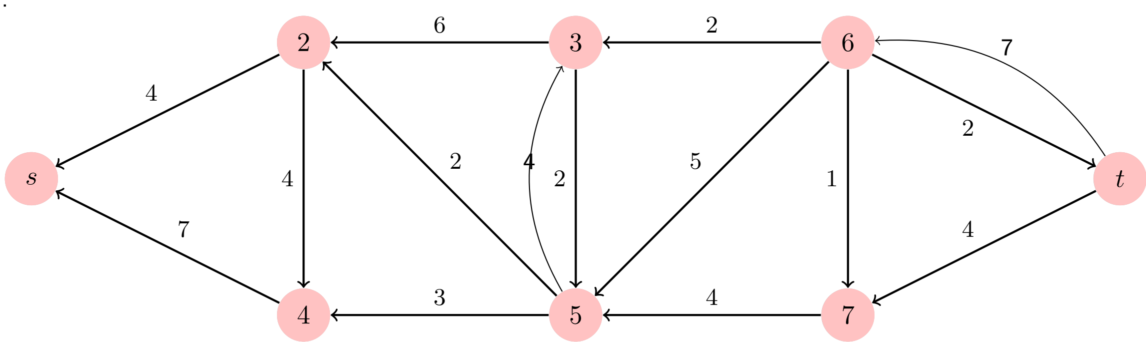
Flow value = 7

Now, from source we have only one option left to choose for node which is node 4 and from node 4 we can only choose path towards node 2. So, select path s -> 4 -> 2 -> 3 -> 6 -> t. We can send 2 units. So the graph becomes-



Flow value = 9

Select path s -> 4 -> 2 -> 5 -> 6 -> t. Almost every node except 6-t can carry 2 units maximum. So, graph -



Flow value = 11.

This is the final residual graph and the max flow of the graph is 11 according to the calculation above. Therefore, it is safe to install a modern shower without causing the historic bathtub to overflow as long as the total flow through the network does not exceed 11 units.

2 Problem 2 Solution :

The choice of algorithm depends on the characteristics of the graph. Most implementations of Ford-Fulkerson take a greedy approach for selecting the augmenting paths. A greedy algorithm always chooses the locally optimal choice at each step, with the hope of finding a global optimum. For selecting augmenting paths, greedy approach can be applied. It can select highest capacity augmenting paths or the paths with shortest number of edges at each iteration.

By using a greedy approach, we can reduce the number of iterations required to find the maximum flow. The idea is to select the augmenting path that maximizes the bottleneck capacity or minimizes the path length, which allows us to increase the flow as much as possible in a single iteration. This can help to reduce the overall computational complexity of the algorithm and make it more efficient.

To implement this approach, we start by selecting an initial augmenting path. Then, at each iteration, we choose the path using greedy method, for example, path with the highest capacity from the source to the sink in the residual graph. We repeat this process until there are no more augmenting paths in the residual graph. The advantage of this approach is that it is simple to implement and does not require additional graph traversal algorithms like BFS or DFS. The greedy approach will select the augmenting path that has the highest capacity bottleneck among all possible augmenting paths. The bottleneck capacity of an augmenting path determines how much flow can be added to the network, and choosing the path with the highest bottleneck capacity ensures that the maximum amount of flow can be added at each iteration. This approach guarantees that the algorithm converges to the maximum flow in a finite number of iterations, even if it might not necessarily be the fastest approach. Following is the process for greedy approach -

1. Initialize the flow to 0 and the residual graph to be the same as the original graph.
2. Select an arbitrary source and sink node.
3. While there exists an augmenting path in the residual graph -
   (a) Find an augmenting path using a greedy approach.
   (b) Find the bottleneck capacity of the augmenting path.
   (c) Update the flow along the augmenting path by adding the bottleneck capacity to the flow of each edge in the path.
   (d) Update the residual graph by subtracting the bottleneck capacity from each forward edge in the path and adding it to each reverse edge.
4. The maximum flow is the total flow leaving the source node.

Here, the key step is the greedy selection of the augmenting path in Step 3a. One common approach is to use the shortest path in terms of the number of edges in the residual graph. Another approach is to choose the path with the largest bottleneck capacity. The choice of the greedy approach can affect the efficiency of the algorithm and the number of iterations required to reach the maximum flow.

The greedy approach for selecting augmenting paths can be justified on the basis that it can improve the efficiency without sacrificing its accuracy and it can lead to a faster convergence to the maximum flow.