**COS 226 – Data Structures and Algorithms**

**Fall 2014 – Flipped Lecture Section**

**Individual/small group worksheet**

**Week 9 – 11.11.14**

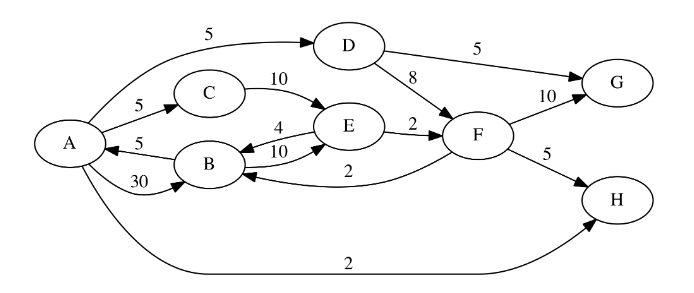
**Topics covered: Shortest Path, maxflow-mincut**

**Solutions**

**Instructions:** This worksheet covers shortest paths (digraphs) and maxflow-mincut problems. Read the worksheet first (before viewing the videos) and understand what type of questions needs to be answered. As you watch videos, if you find the answer to a problem, write the answer here and if possible in salon, so you can share it with others. Also be sure to make some comments/questions on salon.

1. **Dijkstra’s Algorithm**

Consider the graph given below.



1. Find the shortest path from A to F using Dijkstra’s algorithm. Show the PQ and fill in the following table until the shortest path from A to F is found.

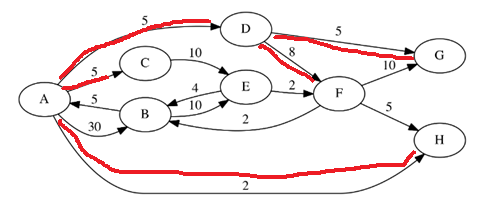
ANSWER

|  |  |  |
| --- | --- | --- |
| Vertex | distTo[ ] | edgeTo[ ] |
| **A** | **0** | **null** |
| B | 15 | F |
| **C** | **5** | **A** |
| **D** | **5** | **A** |
| **H** | **2** | **A** |
| E | 15 | C |
| **G** | **10** | **D** |
| **F** | **13** | **D** |
|  |  |  |
|  |  |  |
|  |  |  |

1. How many other shortest paths were determined in the process of finding the shortest path from A to F? State the vertices and the shortest path edges from A to those vertices.

A🡺 C A🡺 G A🡺 D A🡺F A🡺H

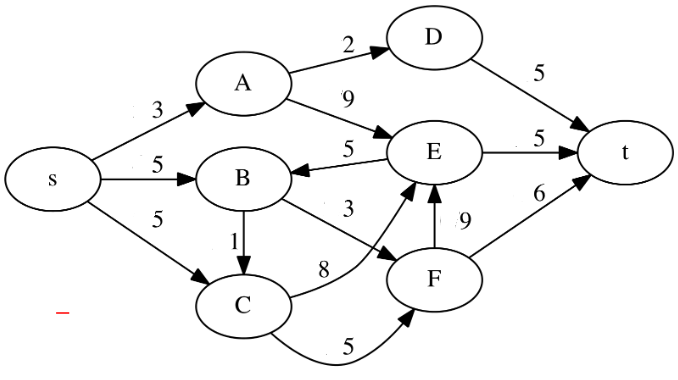
1. Darken the edges in the shortest paths found in the graph. What data structure emerge as a result.



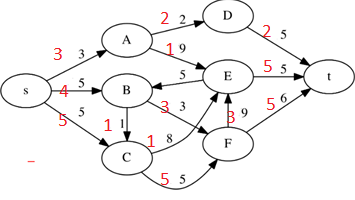
It is a tree

1. **Maxflow-mincut theorem**

Consider the flow diagram given below.



1. Apply the Ford-Fulkerson algorithm to find a maxflow-mincut of the flow diagram. The max capacity for each edge is as given. The max flow is shown in red



1. What is the maxflow of the flow diagram?

Max flow = 12

1. What vertices are included in the mincut that contains the vertex t?

To find the min-cut, start from vertex s and traverse the forward edges that are NOT FULL and backward edges that are NOT EMPTY. In the above graph in part(a), we start with S and traverse to B (only forward edge that is not full).

That is all we can do and we declare the following as the min-cut

s-component ={S,B} t-component = {A,C,E,D,F,t}

Note that all edges out of s-component are full and all edges into s-component are empty.