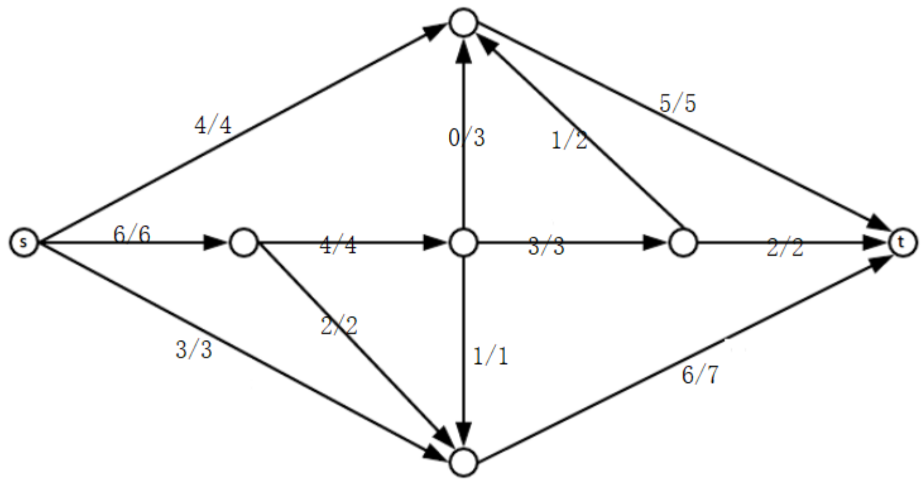
HW8

2018/11/19

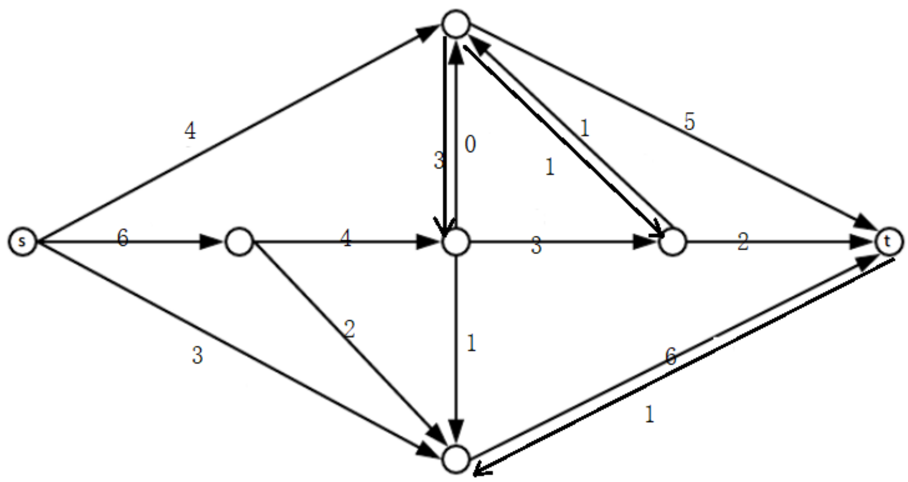
1. As can be seen from the network graph, there are seven paths from s to t.



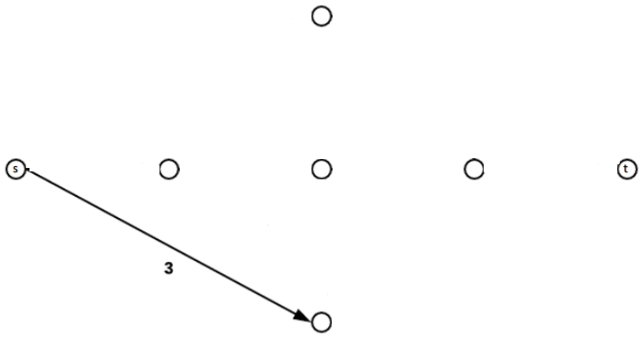
So the max s-t flow is: 5 + 2 + 6 = 13

And according to theory, the minimum s-t cut =5 + 2 + 6 = 13

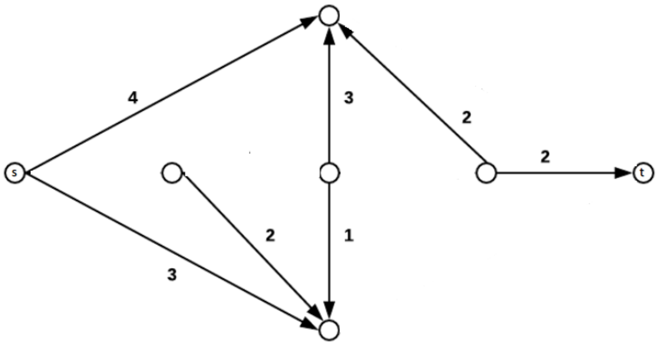
At last, the residual network is following:



2. The upper-binding edges of the network flow are:



The lower-binding edges of the network flow are:



3. Following is the upper-binding-edge-finding algorithm:
   According to Ford-Fulkerson Algorithm, as to the network G, when we get the max s-t flow, there are not any path from s to t in residual network. First we make a Depth-First-Search from s to find reachable vertices in the residual network, and mark them S' set. Then we make a Depth-First-Search from t to find reachable vertices in the reversed residual network, and mark them T' set, the edges that can connect the two point sets (S' and T') is upper-binding edges.

   Time complexity analysis: As we know that when traversing a graph, each point will call DFS function 1 time at most. So the procedure to traversing a graph is a procedure to search the neighbor point of the cur-point. We can reach the conclusion that the time complexity of DFS depends on the data-storage structure it takes. When we choose Adjacency-Table as the data-storage structure, the time complexity is O (m + n). So the time complexity of my algorithm is O (m + n).

   Correctness analysis: It's obviously that if we increase weight of the edges we chose, when we use Ford-Fulkerson Algorithm, these edges will be added in the residual network and the max-flow will be changed. But if we increase other weight-heavier edge, we will not calculate them in the max-flow and the result won't change. So my algorithm is correct.