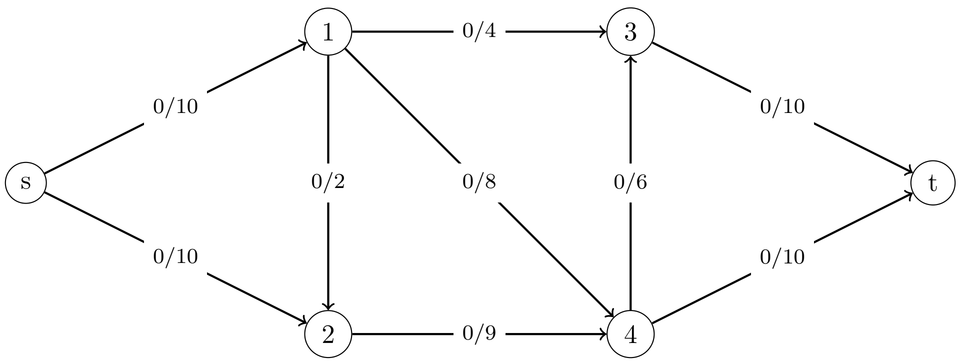
Algorithm Homework 3

龙肖灵
Xiaoling Long
Student ID.:81943968
email:longxl@shanghaitech.edu.cn

November 1, 2017

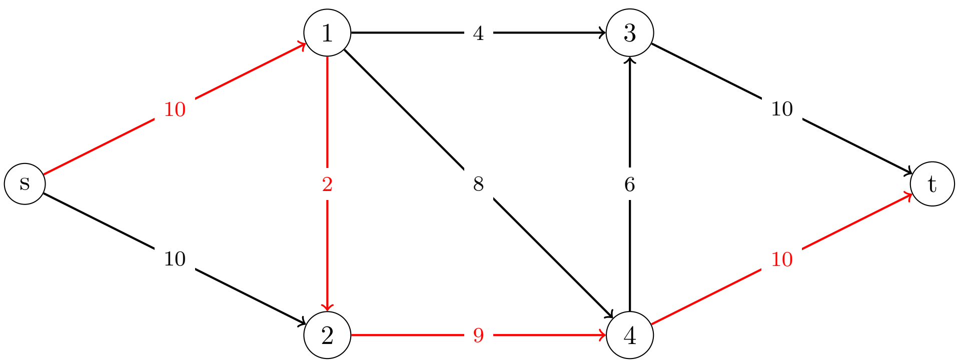
Problem 1

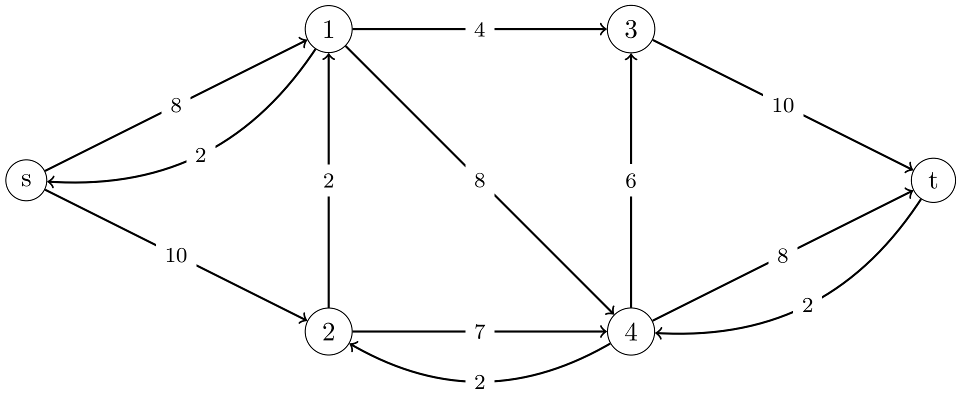
Run the Ford-Fulkerson algorithm on the flow network in the figure below, and show the residual network after each flow augmentation. For each iteration, pick the augmenting path that is lexicographically smallest. (e.g., if you have two augmenting path $1 \to 3 \to t$ and1 $\to 4 \to t$, then you should choose $1 \to 3 \to t$, which is lexicographically smaller than $1 \to 4 \to t$).



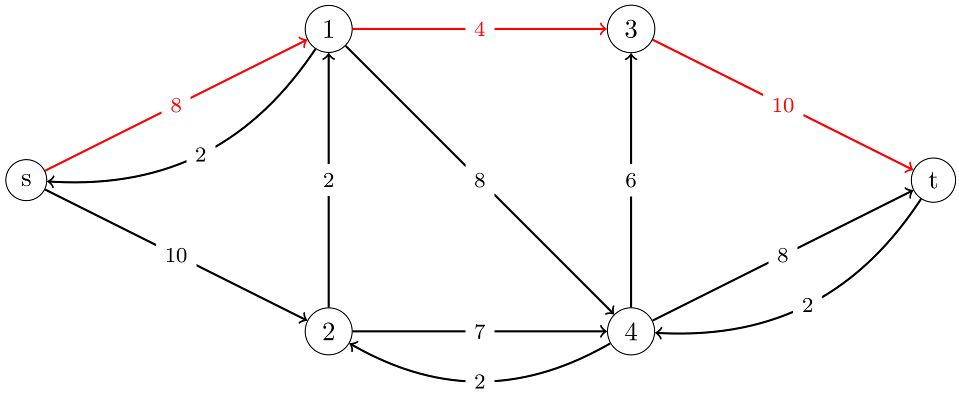
Solution. Suppose that t is bigger than any other number lexicographically.

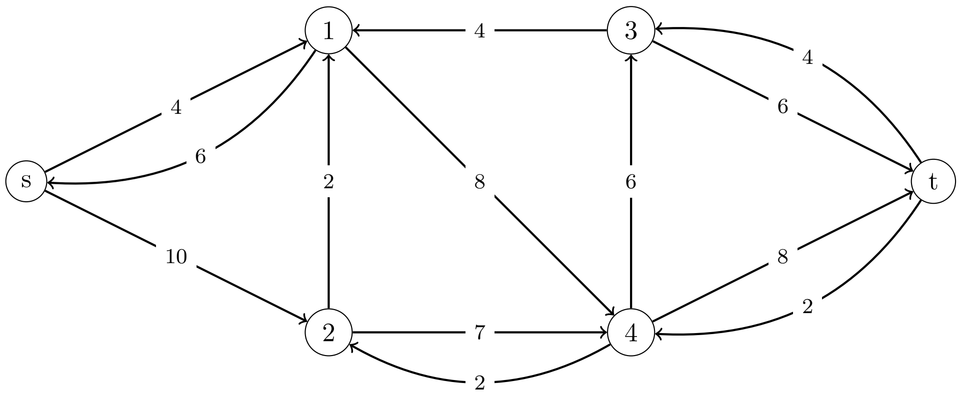
1) Select path $s \to 1 \to 2 \to 4 \to t$. And augment $flow = 2$. Update graph.



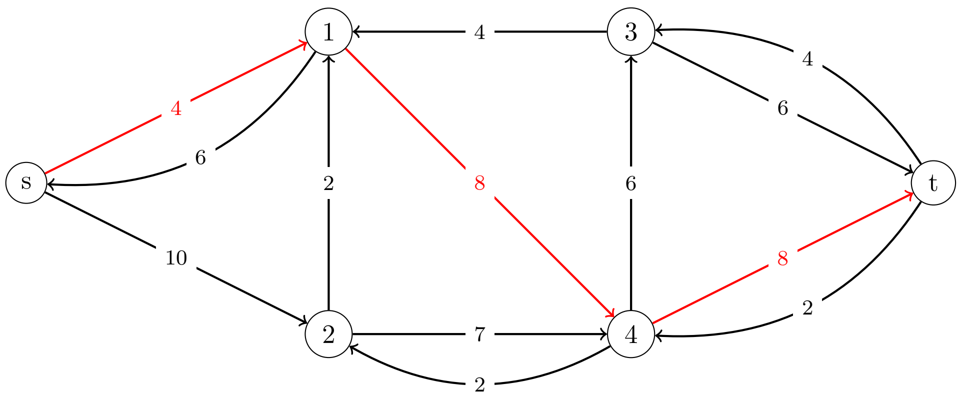


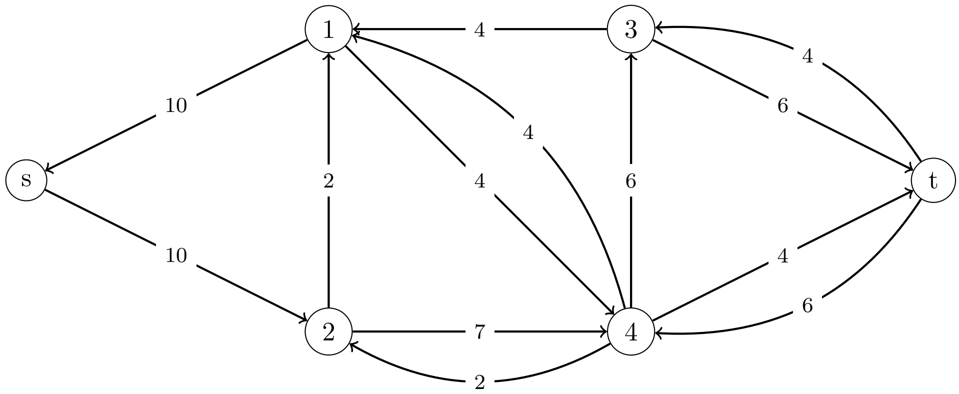
2) Select path $s \to 1 \to 3 \to t$. And augment $flow = 2 + 4 = 6$. Update graph.



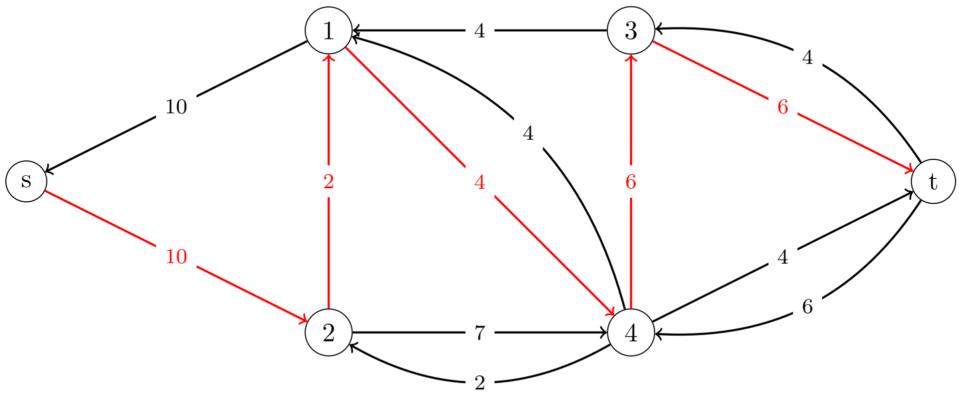


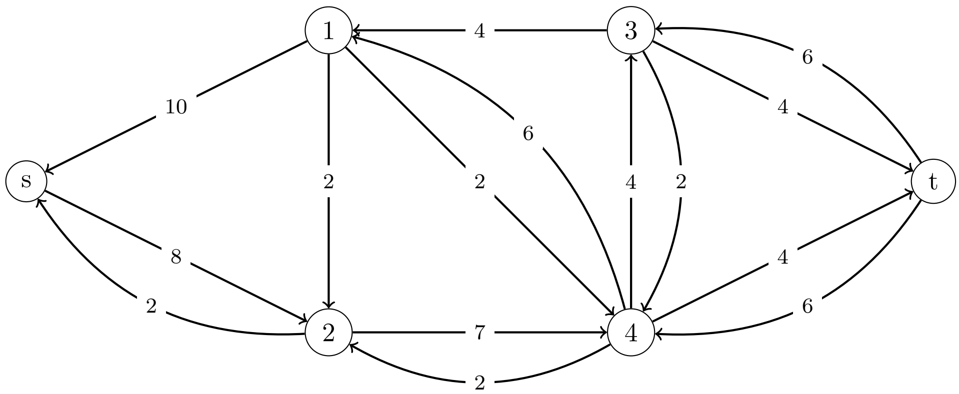
3) Select path $s \to 1 \to 4 \to t$. And augment $flow = 6 + 4 = 10$. Update graph.



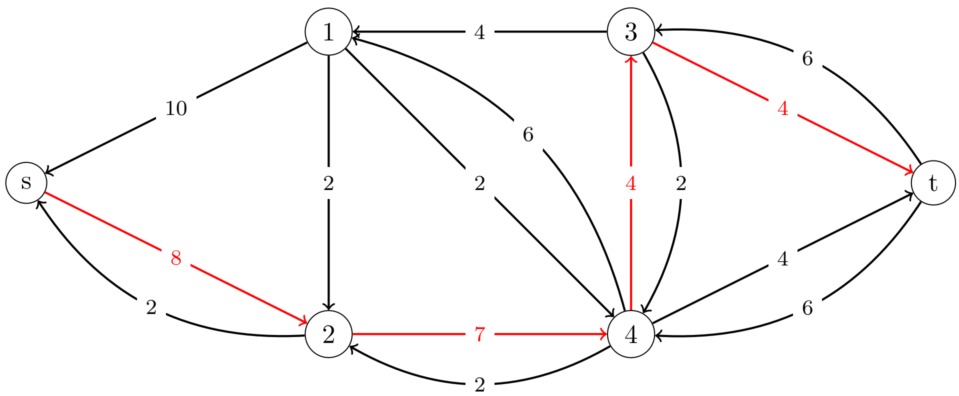


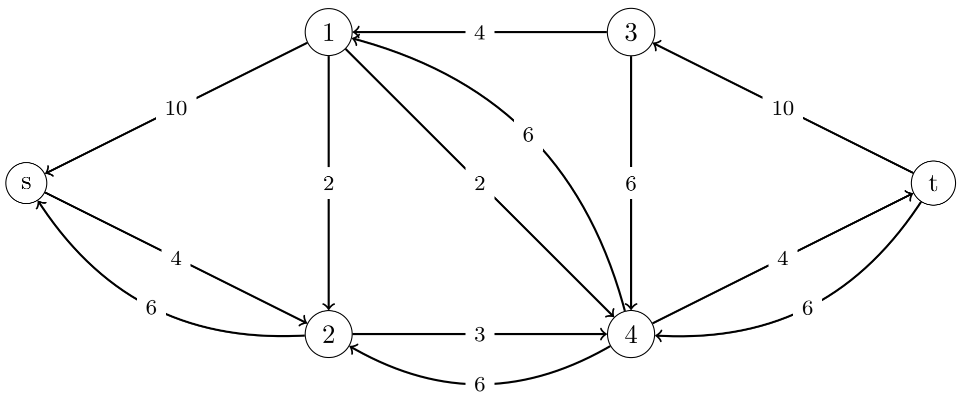
4) Select path $s \to 2 \to 1 \to 4 \to 3 \to t$. And augment $flow = 10 + 2 = 12$. Update graph.



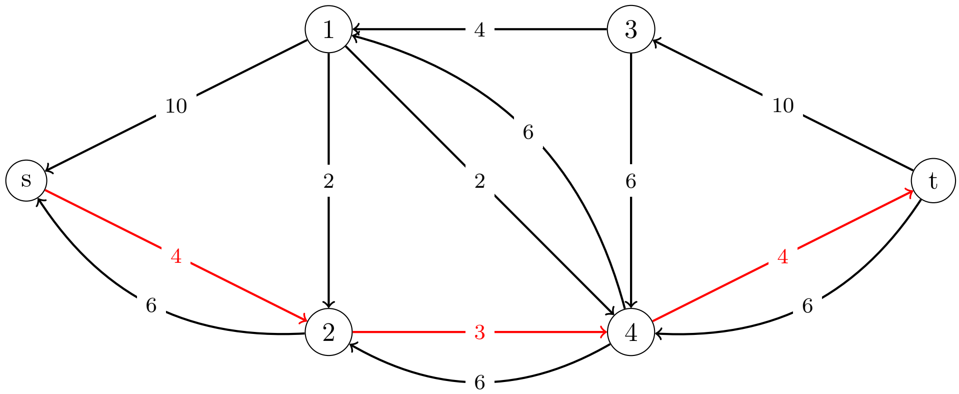


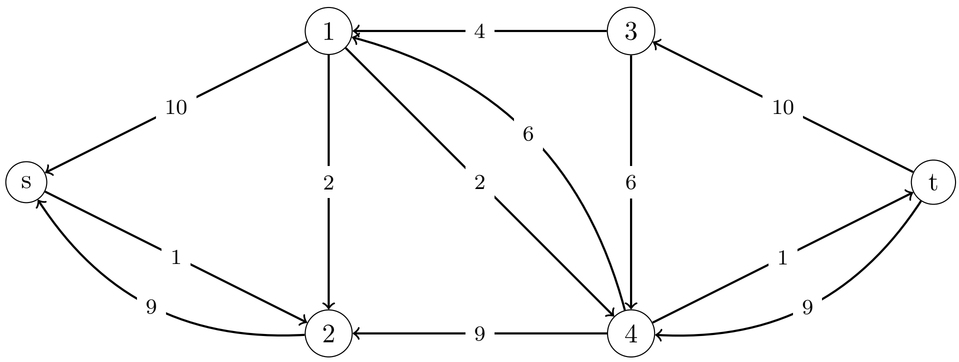
5) Select path $s \to 2 \to 4 \to 3 \to t$. And augment $flow = 12 + 4 = 16$. Update graph.





6) Select path $s \to 2 \to 4 \to t$. And augment $flow = 16 + 3 = 19$. Update graph.





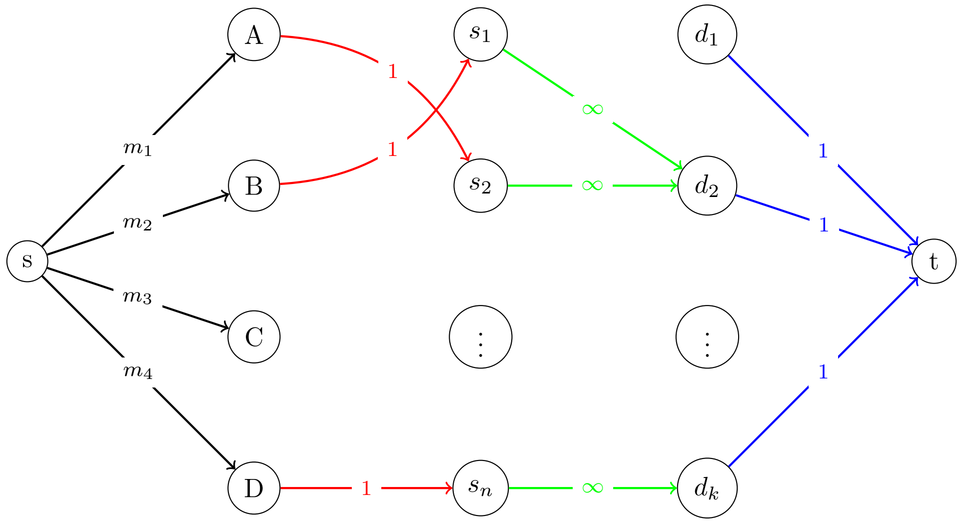
7) Finally, there is no path now, we can get maximum flow is 19.

Done. ■

Problem 2

There are n staff members in a company. Each staff member belongs to a department. There are k departments. Now we have to choose one staff member from each department to a company-level committee. In this committee, there must be m_1 number of A-class staff members, m_2 number of B-class staff members, m_3 number of C-class staff members, and m_4 number of D-class staff members (so we have $m_1 + m_2 + m_3 = m_4 = k$). We are given the list of staff members, their home departments, and their classes (A, B, C, D). Describe an efficient algorithm to determine who should be on the committee such that the constraints are satisfied.

Solution. We can regard this problem as a maximum flow problem after do some modification as bellow.



(a) Black lines: we need select m_i ith-class staff number(s).

(b) Red lines: s_i belongs to ith-class

(c) Green lines: s_i lives in d_i department.

(d) Blue lines: every department we need a commitee.

So the problem become find the maximum flow in above graph. So we can solve this problme by find the maximum in above graph. If the maximum flow is k, then we can get the committee member satisfying the constraints. And if not, we cannot satisfy all constraints. And All staff member s_i in maximum flow will be on the committee.
Done. ■

Problem 3

Let $G = (V, E)$ be a flow network with source s, sink t, and integer capacities. Suppose that we are given a maximum flow in G.

1. Suppose that we increase the capacity of a single edge $(v, u) \in E$ by 1. Give an $O(V + E)$ time algorithm to update the maximum flow.

2. Suppose that we decrease the capacity of a single edge $(v, u) \in E$ by 1. Give an $O(V + E)$ time algorithm to update the maximum flow.

Solution.

1. Add this capacity to the last residual graph. Do a BFS on this residual network from s. If find a path from s to t, then maximum flow add 1, and the new maximum flow will augment by 1 following this path the flow value increased by 1, or not maximum flow remains. And the run time of DFS is $O(V + E)$. It can satisfy the constraint.

2. Asume that maximum flow has no circle.(Find a maximum flow without choosing a circle).

   a. Do a BFS on the last residual graph to find a path from t to u, decrease the flow of the edge on path by 1. Update the residual graph.

b. Do a BFS on the last residual graph to find a paht from v to s, decrease the flow of the edge on path by 1. Update the residual graph.

c. Decrease the flow of the edge $(v, u) \in E$ by 1. Just do $f(e^R) = f(e^R) - 1$ on new residula graph.

d. Do a BFS on the new residual graph. If find a path from s to t, then update the maximum flow. The flow value remains. If not, maximum flow is on the new residual graph. And the maximum value decreased by 1.

e. The run time of this algorithm can be $O(V + E)$, since we just do some BFS.

Done. ■

Problem 4

Suppose we have an n-by-m matrix in which each element is either 0 or 1. We know the row sums $r = (r_1, \cdots, r_n)$ and the column sums $c = (c_1, \cdots, c_m)$ (it is guaranteed that $\sum_i r_i = \sum_j c_j$). Design a polynomial-time algorithm based network flow to decide whether such a matrix exists or not. Examples: $n = m = 3$ with $c = (1, 2, 0)$ and $r = (1, 1, 1)$ (answer=yes) or $r = (3, 0, 0)$ (answer=no):

Yes

1	0	0	1
0	1	0	1
0	1	0	1
1	2	0	

No

1	1	1	3
			0
			0
1	2	0	

Solution. To built a network based on network flow, and we can decide whether there exists a matrix or not in polynomial-time. Divide into 5 layers.

1) layer 1: Source node s.

2) layer 2: n row sums R_i. And edge $c(s - R_i) = r_i$.

3) layer 3: $n \times m$ nodes of elements from matrix. And for all j from 1 to m, edge $c(R_i - m_{i,j}) = 1$.

4) layer 4: m column sums C_j. And for all i from 1 to n, edge $c(m_{i,j} - C_j) = 1$.

5) layer 5: Sink node t. Edge $c(C_j - t) = c_j$.

Then solve this maximum flow problem. If we can find a maximum flow that $flowvalue = \sum_i r_i = \sum_j c_j$, then the matrix exists. And the elements go through will be 1, and others be 0. Or not there doesn't exist such a matrix.
Done. ■

Problem 5

Consider a network $G = (V, E)$. In addition to having a capacity $c(u, v)$ for every edge (u, v), we also have a capacity $c(u)$ for every vertex u. A flow f is feasible only if, in addition to the conservation constraint and the edge capacity constraint, it also satisfies the vertex capacity constraint: $\forall v \in V, \sum_{u:(u,v)\in E} f(u, v) \le c(v)$. Give an efficient algorithm to solve the maximum flow problem on this network.
Hint: You can reduce the problem to the standard maximum flow problem.

Solution. We can reduce the problem to the standard maximum flow problem after process as bellow.

(a) Assume there exist a source node and a sink node.

(b) Split every node u into 2 node u_1 and u_2 except source node and sink node. And there is an edge $c(u_1, u_2) = c(u)$.

(c) All edge (v, u) become (v, u_1) and the capacity of edge (v, u_1) equals to (v, u).

(d) All edge (u, v) become (u_2, v) and the capacity of edge (u_2, v) equals to (u, v).

(e) Then the origin problem can be solved as a standard maximum flow problem.

Done. ■

Problem 6

Given a network $G = (V, E)$, design a polynomial time algorithm to determine whether G has a unique minimum $s - t$ cut.

Solution. Inspirated by Problem 3. First find the minimum $s - t$ cut of G, and the edges between cut. Then for each e between cut, increase the capacity of it. If there is an edge whose maximum flow in new network graph doesn't change, then the miimum cut isn't unique. Or all edge increasing change the maximum flow, means the minimum cut is unique.
Actually, if an edge between cut increased by 1, and the maximum flow doesn't increase. That means the child node of this node doesn't have more capacity for flow. So this child node can get into minimum cut. The minimum cut isn't unique.
Done. ■