1 Notations

1. $\delta^+(v)$ is the set of arcs that are coming out of v, $\delta^-(v)$ is the set of arcs that are coming into the vertex v.
2. $\mathbb{1}_M$ is the indicator vector of some sets. In the case of graph, let M be a set of edges, then $\mathbb{1}_M \in \mathbb{R}^{|E|}$ and $(\mathbb{1}_M)_e = 1$ if $e \in M$ else it's just zero.

2 Problem 4.10

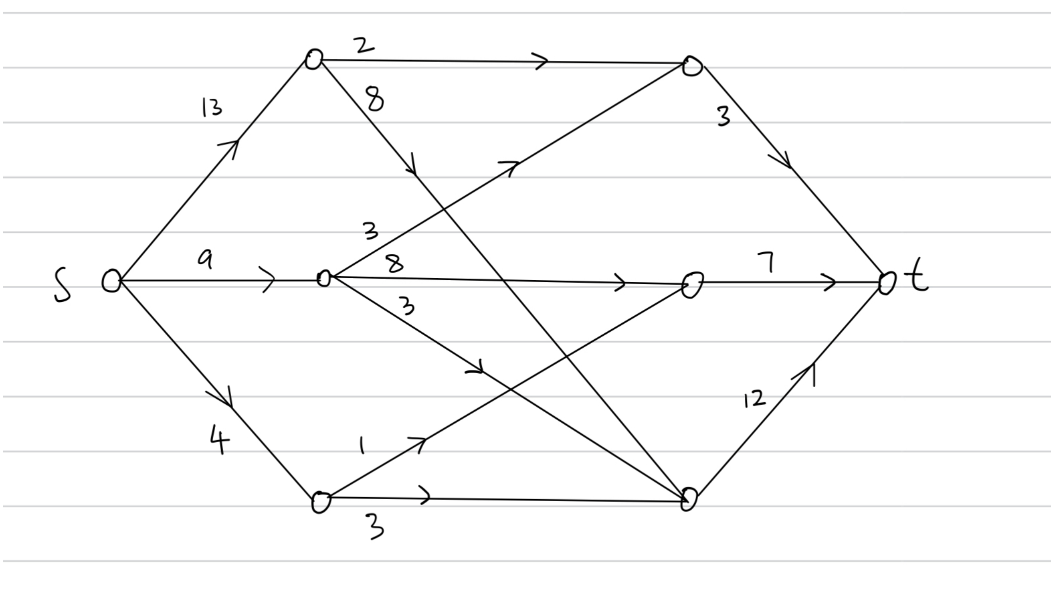
Determine using the max flow algorithm if thre exists a 3×3 matrix whose row sum is $\le [13, 9, 4]$ and whose column sum is $= [3, 7, 12]$ and the matrix is:

$$A \le \begin{bmatrix} 2 & 0 & 8 \\ 3 & 8 & 3 \\ 0 & 1 & 3 \end{bmatrix} \tag{2.0.1}$$

2.1 Reduction Strategies

There are 3 sets of inequalities, each corresponds to the capacity limit of each edges on the graph, in addition. The summation happens at the vertices.

Entries of the matrix can be modeled by edges going between 2 sets of bipartite graph. On each side of the bipartite graph we connect the s, t vertices, and we link s, t to each vertices in each group by one edge, and those edges corresponds to the row sum and the column sum. The matrix A exists iff there is a way to send flow on the above graph such that the value



of the flow equals to $3 + 7 + 12$, saturating all the edges coming into the vertex t.

2.2 Applying Network Algorithm

Actually solving the flow by hands:

§ HW7 4.10 is like:

$a_1 + a_2 + a_3 \le \begin{bmatrix} 13 \\ 9 \\ 4 \end{bmatrix}$

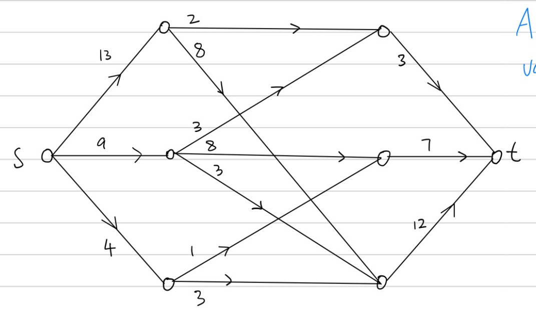
$A = \begin{bmatrix} | & | & | \\ a_1 & a_2 & a_3 \\ | & | & | \end{bmatrix}$

$\langle \mathbb{1}, a_1 \rangle = 3$, $\langle \mathbb{1}, a_2 \rangle = 7$

$\langle \mathbb{1}, a_3 \rangle = 12$

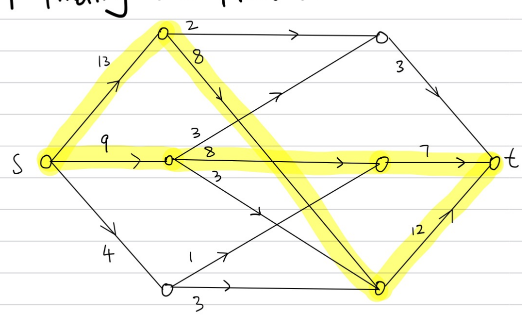
$a_1 \le \begin{bmatrix} 2 \\ 3 \\ 0 \end{bmatrix}$ $a_2 \le \begin{bmatrix} 0 \\ 8 \\ 1 \end{bmatrix}$ $a_3 \le \begin{bmatrix} 8 \\ 3 \\ 3 \end{bmatrix}$

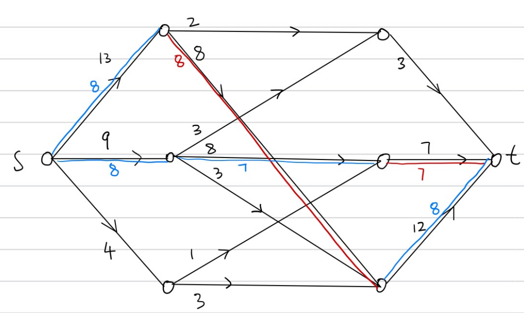
$0 \le A \le \begin{bmatrix} 2 & 0 & 8 \\ 3 & 8 & 3 \\ 0 & 1 & 3 \end{bmatrix}$

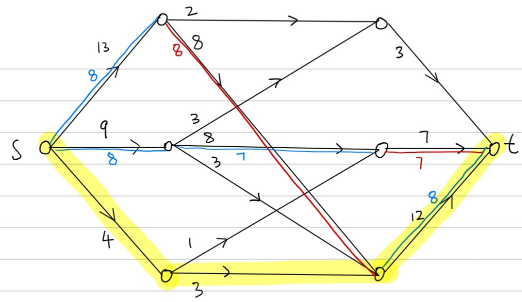


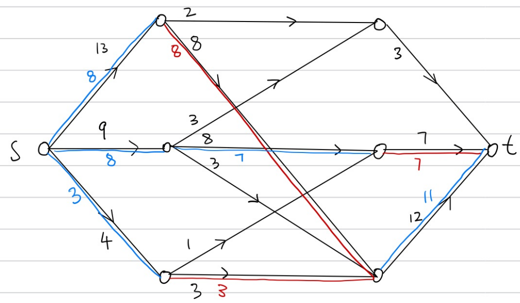
A exists if a flow value of 22 exists.

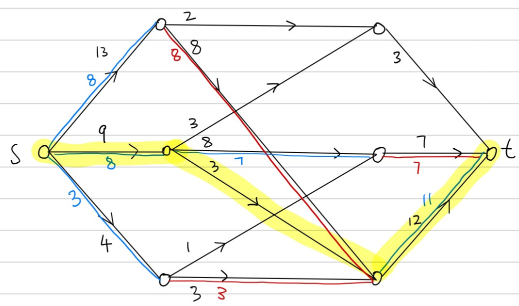
Let's start finding the flow !!!!

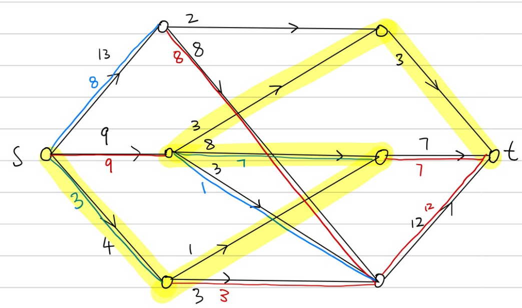


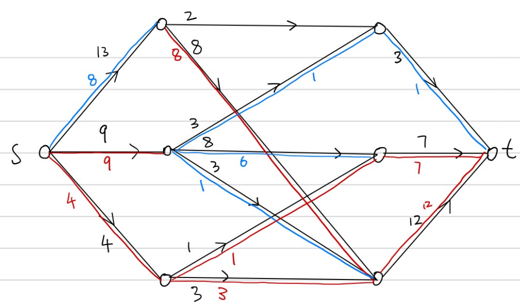


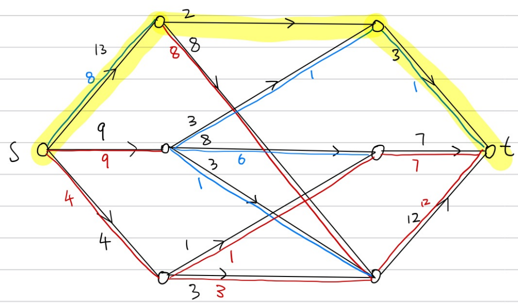

s
t
2
13
8
8
8
3
9
8
3
8
3
7
7
7
4
1
3
12
8

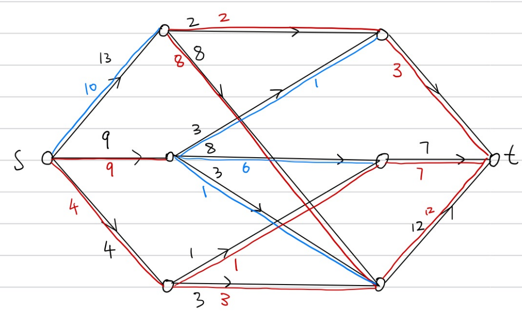

s
t
2
13
8
8
8
3
9
8
3
8
3
7
7
7
3
4
1
3
3
12
11


s
t
2
13
8
8
8
3
9
8
3
8
3
7
7
7
3
4
1
3
3
12
11


s
t
2
13
8
8
8
3
9
9
3
8
3
7
7
7
1
3
4
1
3
3
12
12







*flow possible

$$A = \begin{bmatrix} 2 & 0 & 8 \\ 1 & 6 & 1 \\ 0 & 1 & 3 \end{bmatrix} \quad A\mathbb{1} = \begin{bmatrix} 10 \\ 8 \\ 4 \end{bmatrix} \quad \mathbb{1}^T A = \begin{bmatrix} 3 & 7 & 12 \end{bmatrix}$$

3 Problem 4.15

Proposition 3.1. Let $D = (V, A)$ be a directed graph, and let $f : A \mapsto \mathbb{R}_+$, let $\mathcal{C}$ be the collection of directed circuits in D. For each directed circuit C in D, let χ^C be the incidence vector of C, that is, $\chi^C : A \mapsto \{0, 1\}$, $(\chi^C)_a = 1 = \chi^C(a)$ if C traverse the arc a else $\chi^C(a) = 0$.

f is a non-negative circulations if and only if there exists a function $\lambda : \mathcal{C} \to \mathbb{R}_+$ such that:

$$f = \sum_{C\in\mathcal{C}} \lambda(C)\chi^C \tag{3.0.1}$$

Simply put, all non-negative flow on the graph form the cone of all circuits vector on the graph.

3.1 Proof of Necessity $\Longleftarrow$

We wish to show that the cone of circuits indicators vectors are still a non-negative flow. To characterize non-negative flow on the digraph, we define the incidence matrix M using the following:

$$M := \begin{bmatrix} \mathbb{1}^T_{\delta^+(v_1)} - \mathbb{1}^T_{\delta^-(v_1)} \\ \mathbb{1}^T_{\delta^+(v_2)} - \mathbb{1}^T_{\delta^-(v_2)} \\ \vdots \\ \mathbb{1}^T_{\delta^+(v_n)} - \mathbb{1}^T_{\delta^-(v_n)} \end{bmatrix} \tag{3.1.1}$$

Then the polyhedron $P := \{x \geq \mathbf{0} | Mx = \mathbf{0}\}$ will characterize all the non-negative flow on the graph (This is directly by definition of non-negative flow). Consider any $\chi^C \geq \mathbf{0}$, which is non-negative, there sum will be non-negative too.

C is a circuit, therefore $\forall v \in C : \delta^+(v) = \delta^-(v)$, therefore the incidence vector χ^C has property $M\chi^C = \mathbf{0}$. Therefore:

$$M \sum_{c\in\mathcal{C}} \lambda(C)\chi^C = \sum_{c\in\mathcal{C}} \lambda(C)M\chi^C = \mathbf{0} \tag{3.1.2}$$

$$\implies \sum_{c\in C} \lambda(C)\chi^C \in P \tag{3.1.3}$$

Therefore, for all element in the cone spanned by the incidence vector of all cirtuits on the graph, it's also in the polyhedron that characterize non-negative flow on the graph.

3.2 Proof of Sufficiency $\implies$

Given any non-negative flow, it can be represented by some element in the cone cone($\mathcal{C}$). If the non-negative circulation is all zero, then it's trivialy true, just make $\lambda(C) = 0 \; \forall \; C$.

Next, assuming that the circulation $f^{(0)} = f$ is strictly positive, we use $\chi^{f^{(0)}}$ as an indicator vector to denote all the arc in the circulation, then there must exist χ^C for some $C \in \mathcal{C}$ such that for all arcs in C, it's in $f^{(0)}$. For contradiction if this is not true, then there exist some path where the flow ends at a vertex v, making $\sum_{a\in\delta^+(v)} f(a) \neq \sum_{a\in\delta^-(v)} f(a)$, hence f is not a circulation.

Let C_0 be the circuit such that for all $a \in C_0 : f(a) \geq 0$. Choose $0 < \alpha_1 = \min_{a\in C_0} f(a)$, then:

$$f^{(1)} := f^{(0)} - \alpha_1 \chi^{C_0} \tag{3.2.1}$$

$f^{(0)}$ is still a circulation. This is true because forall $v \in V$, it's either v is covered by circulation $f^{(0)}$, or it's covered by $f^{(0)} \cup C_0$. In the former case, the flow coming in and out is equal, in the latter case:

$$\sum_{a\in\delta^+(v)\cap C_0} f^{(0)}(a) = \sum_{a\in\delta^-(v)\cap C_0} f^{(0)}(a) \tag{3.2.2}$$

$$\implies \sum_{a\in\delta^+(v)} f^{(0)}(a) - \sum_{a\in\delta^+(v)\cap C_0} \alpha_1 f^0(a) = \sum_{a\in\delta^-(v)} f^{(0)}(a) - \sum_{a\in\delta^-(v)\cap C_0} \alpha_1 f^{(0)}(a) \tag{3.2.3}$$

Therefore $f^{(1)}$ is still a circulation. Let's consider the statement inductively, then one can show that:

$$\mathbf{0} < f^{(1)} := f^{(0)} - \alpha_1\chi^{C_0} \quad \alpha_1 := \min_{a\in C_0} f^{(0)}(a) \tag{3.2.4}$$

$$\mathbf{0} < f^{(2)} := f^{(1)} - \alpha_2\chi^{C_0} \quad \alpha_2 := \min_{a\in C_1} f^{(1)}(a) \tag{3.2.5}$$

$$\vdots \tag{3.2.6}$$

$$\mathbf{0} = f^{(n)} := f^{(n-1)} - \alpha_{n-1}\chi^{C_{n-1}} \quad \alpha_{n-1} := \min_{a\in C_{n-1}} f^{(n-1)}(a) \tag{3.2.7}$$

The sequence of non-negative circulation will terminates with exactly zero. This is inductively true because all circulation must contain a circuit, if it's not, then it's not a circulation. Therefore, the original initial circulation can be expressed in the form of:

$$\mathbf{0} = f^{(n-1)} - \alpha_{n-1}\chi^{C_{n-1}} \tag{3.2.8}$$

$$\mathbf{0} = f^{(n-2)} - \alpha_{n-2}\chi^{C_{n-2}} - \alpha_{n-1}\chi^{C_{n-1}} \tag{3.2.9}$$

$$\vdots \tag{3.2.10}$$

$$\mathbf{0} = f^{(0)} - \left(\sum_{j=1}^{n-1} \alpha_j \chi^{C_j}\right) \tag{3.2.11}$$

And we have expresed the non-negative circulation $f^{(0)}$ as a non-negative summation of all the indicator vector for circuits on D. All citcuits on D is coming from $\mathcal{C}$, therefore, all non-negative circulations are inside of cone($\mathcal{C}$).

4 Problem 4.18

Describe the problem of finding a max-weighted matching in a bipartite graph as a min-cost flow problem.

4.1 Some Preparations for the Proofs

Lemma 4.1.1 (Lemma 1)**.** Given a bipartite graph $G := (V\dot{\cup}U, E)$, the maximum weight matching of a $w : E \mapsto \mathbb{R}_+$has to be a maximum cadinality matching first.

Proof. For contradiction assuming that M is not a maximum cardinality matching and it has maximum weight. Choose M^+ that is a maximum cardinality matching, then the set $M^+\Delta M$ is non zero, and there exists edge $e \in M^+$ but $e \notin M$. Choosing such an edge and include it to M gives more weight for M because the weights are non-negative. Contradiction is shown. □

We make the notation of converting an arc $a = (u, v)$ into an edge as $E(a) = \{u, v\}$.

The reduction process goes by converting the undirection Bipartite graph into a directed graph, and then convert the weight function into a cost function, and then define the fix amount of flow for the min cost flow algorithm.

4.2 The Reduction Process

Given bipartite graph $G := (U\dot{\cup}V, E)$, construct:

$$A = \{(u, v)|u \in U \wedge v \in V\} \tag{4.2.1}$$

$$D := (\{s, t\} \cup U\dot{\cup}V, A \cup \{(s, u)|u \in V\} \cup \{(v, t)|v \in V\}) \tag{4.2.2}$$

$$c : E \mapsto \mathbb{R}_+ \equiv 1 \tag{4.2.3}$$

The amount of flow that we wish to send equals to $|M|$ where M is the maximum cardinality matching on G. All edges are having a capacity of 1, therefore there exists maximum value flow solution as an integral flow on the graph D.

Observe that for all $v \in V$, $\sum_{a\in\delta^+(v)} f(a) \leq 1$, and similarly for all $u \in U$, $\sum_{a\in\delta^-(v)} f(u) \leq 1$ as well, which implies that all integral maximum flow f on D is a matching, in addition the arcs that is in f and goes between the set U, V will be a maximum cardinality matching between U, V. This is true because the value can be computed as:

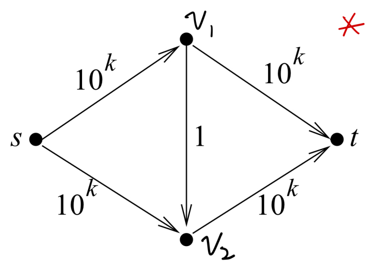
$$\text{val}(f) = \sum_{a\in\delta^+(s)} f(a) = \sum_{v\in V} \sum_{a\in\delta^-(v)} f(a) = |M| \tag{4.2.4}$$

Next, define the cost function $k(a) = -w(E(a))$, the cost function is just the negative weight. Since w is $\mathbb{R}_+$, then k is $\mathbb{R}_-$. Minimizing the cost among all flow with a flow of $|M|$ will correspond to maximumizing the weight for all matching on G with cardinality $|M|$. The solution with maximum weight must has a flow value the same as a matching with max cardinality, this is proved in lemma 4.1.1, therefore, the minimum cost among all flow must be a flow with a value $|M|$.

5 Problem 4

Justify the statement on page 65 of Schrijver's notes that, with a bad choice of flow-augmenting paths, the Ford-Fulkerson uses $2{*}10^k$ iterations of the flow-augmenting algorithm to find an s-t flow of maximal value in the graph Figure 4.3. (To do this, you should give an explicit sequence of flow-augmenting paths.)

The graph from page 65 is:



* paths always go through the arc with capacity 1.

Figure 4.3

One strategy of choosing the sequences of flow-augmenting path is to always choose include the edges with the least capacity. In this case the flow must path through the arcs $(v_1, v_2), (v_2, v_1)$. The sequence of path would be:

$$P_1 = s - v_1 - v_2 - t \tag{5.0.1}$$
$$P_2 = s - v_2 - v_1 - t \tag{5.0.2}$$
$$\vdots \tag{5.0.3}$$
$$P_{2\times 10^k} = s - v_1 - v_2 - t \tag{5.0.4}$$

And each time we send a flow of exactly 1, therefore we need 2×10^k to saturate the arcs in $\delta^-(t)$. The first flow path create $f(a^{-1}) = (v_2, v_2)$ on the residual graph, and the path P_2 set it to zero, and the cycle repeats.