CS5800: Algorithms — Virgil Pavlu

Homework 12

Name: MUMUKSHA PANT
Collaborators:

Instructions:

- Make sure to put your name on the first page. If you are using the LaTeX template we provided, then you can make sure it appears by filling in the `yourname` command.
- Please review the grading policy outlined in the course information page.
- You must also write down with whom you worked on the assignment. If this changes from problem to problem, then you should write down this information separately with each problem.
- Problem numbers (like Exercise 3.1-1) are corresponding to CLRS 3^{rd} edition. While the 2^{nd} edition has similar problems with similar numbers, the actual exercises and their solutions are different, so make sure you are using the 3^{rd} edition.

1. (20 points) Exercise 26.1-3.

Solution:
There is a vertex u for which there is no path from s to t . There is another vertex v where a path exists from $s->v->t$ then f(v,u) and f(u,v) != 0

There can be 2 cases here:
Case 1- there is no path from source (s) to u .
Case 2 - there is no path from u to t (sink)
In both cases, if at any point we have a cycle of vertices, this means we can reduce the flow around cycle by min flow going through it and this will not change value of flow f(s,t) . Flow f(s,t) remains unchanged because it is not appearing in the cycle. The idea here is that any flow $s->u$ must be cyclic and can be made 0 without affecting the overall flow.

Another approach:
Let G be the network flow of such a graph and f be its flow. Then , Since u does not have any path from s to t (source to sink) , we can remove. this edge from G and there will be no effect on our net flow f (because $s->u->t$ no path)
Let G′ be the resulting graph after we have removed u . We are claiming that f is also the max flow in G′ .

Let's prove this by contradiction.
f is the max flow and let there be f′ where $f<f'$
The flow f′ in G′ is f′(u,v) =0 = f′(v,u)
Now when we add G′ back to construct G, the flow will remain unchanged. That is : $f+f'=f$ this means f′ = 0
This proves our point that is there is no path from a vertex from source to sink, the maximum flow from it will be 0.

---

2. (20 points) Exercise 26.1-4.

Solution:
collaborated from Quizlet site.
To prove that flow f1 and f2 forms a convex set, $\alpha f1+(1-\alpha)f2$ ∀ [0 <= α <= 1], it needs to satisfy 2 properties:
1) Capacity constraint
2) Conservation of Flow

26.1-4 f1 & f2 are flows (All 2 properties).

$\alpha f1 + (1-\alpha) f2$

① capacity

$f1(u,v) \le c(u,v)$

$f2(u,v) \le c(u,v)$

// flow through u,v can be lesser or equal to capacity of u,v

$$\alpha f1(u,v) + (1-\alpha) f2(u,v) \le \alpha c(u,v) + (1-\alpha) c(u,v)$$

$$\le c(u,v)[\cancel{\alpha} + 1 - \cancel{\alpha}]$$

$$\le c(u,v)$$

This means capacity constraint is maintained & satisfied.

Quizlet

② Conservation of flow.

$$\sum_{v \in V} f(u,v) = 0 \qquad u = V - \{s,t\}$$

$$\sum_{v \in V} \alpha f1(u,v) + (1-\alpha) f2(u,v)$$

$$= \alpha \sum f1(u,v) + (1-\alpha) \sum f2(u,v)$$

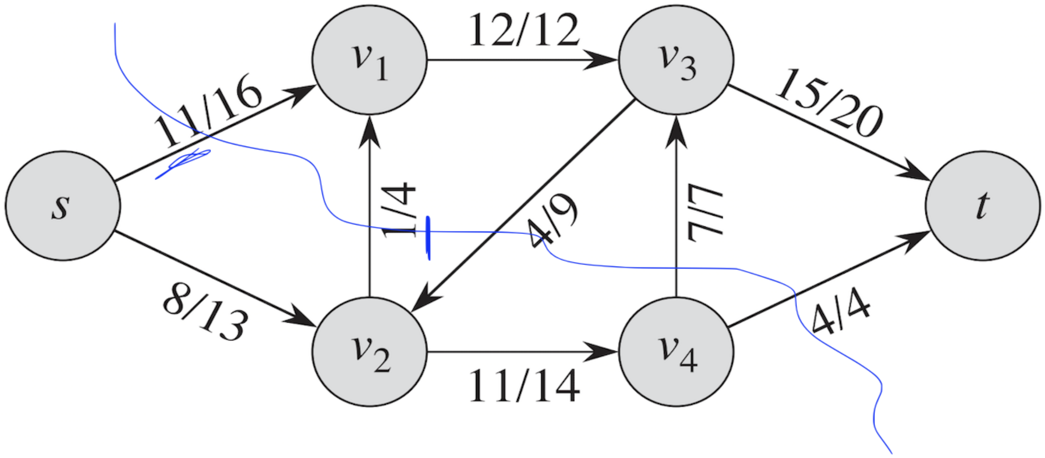
$$\alpha (0) + (1-\alpha)\, 0$$

$$= 0 + 0 = 0$$

---

3. (20 points) Exercise 26.2-2.

Solution:



(b)

Flow across cut:
f(s,v1) : 11
f(v2,v1) :1
f(v2,v3) :-4
f(v4,v3) :7
f(v4,t):4

Flow accross cut is : 19

The capacity of the cut is:
c(s,v1) : 16
c(v2,v1) :4
c(v2,v3) :0
c(v4,v3) :7
c(v4,t):4

capacity is: 16 + 4 + 7 + 4 = 31

---

4. (Extra Credit 20 points) Exercise 26.2-10.

5. (30 points) Implement Push-Relabel for finding maximum flow.
Extra Credit: use relabel-to-front idea from Chapter 26.5 with the Discharge procedure.

Solution:

6. (15 points) Explain in a brief paragraph the following sentence from textbook page 737: "To make the preflow a legal flow, the algorithm then sends the excess collected in the reservoirs of overflowing vertices back to the source by continuing to relabel vertices to above the fixed height $|V|$ of the source".

Solution:

In push relabel algorithm , unlike Ford Fulkerson and other methods, here the conservation law in every vertex is not maintained. Conservation Law state that for every vertex except the source and the sink the total inflow should be equal to the total outflow.

But in Push relabel, this law is not followed and every vertex can have "excess flow" which is stored in the reservoir for every vertex. Reservoirs are used to store "overflow" or "excess flow" for each vertex.

To make this a "legal" flow - that is to ensure that whatever amount inflows from the source(s) the same amount is outflowed through the sink (t) , we send the excess flow back to source if there is no path from the vertex which has the excess flow to the sink (t) .

Sending back to source is done by using relabelling (height increasing)

---

7. (Extra Credit 20 points) Exercise 26.4-4.