PASS Session

Wednesday, 1 June, 2022 21:06

Network Flow

Fundamentals

- Vertex **without incoming edge** → source
- Vertex **without outgoing edge** → target
- Edges → Capacity, how much it can "contain"
- Flow → How much is it "flowing"
- Flow constraint property → Flow **can't be more** than capacity of edge
- Flow conservation property → Except for source and target, incoming flow == outgoing flow

Maximum Flow Problem

1. Find Total Flow → identify total flow from target/source (note: it will always be the same)
2. Want to find maximum flow possible, but risk of Flow Constraint
3. How to tackle this? ***Ford-Fulkerson***

Ford-Fulkerson

- Residual Edge → Remaining capacity of given edge. If flow == capacity, residual is 0
- Reversible Edge → Flow that can be cancelled (a.k.a **usually** your edge flow but in the reverse direction)
- Residual Network → Graph that contains only residual edge, can choose to exclude the 0 residual edges
- We find maximum flow using path augmentation
- Sum if multiple residual/reversible edge pointing to same direction, such sum equals edge capacity

Example

See whiteboard

Complexity

- Initialize Residual Network → O(E)
- Path Augmentation → O(V+E) Since we just do BFS/DFS to find path and augment
- Whole process → O(F) where F is flow itself. Increase flow until unable to
- **Total → O(FE)**
- However, in FIT3155, F can equal to O(VE^2) with Edmonds-Karp (Haven't gone through it yet sry ><)

Cut

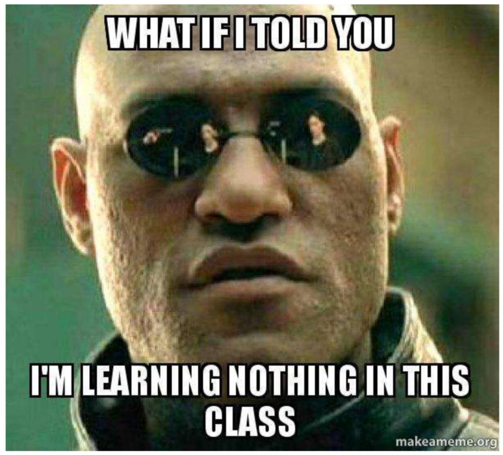
- Cut network flow into two
- Divide into two sections → vertices with s included and vertices with t included
- Capacity of cut → Capacity of outgoing edges from cut
- Flow of cut → Flow of outgoing edges – flow of incoming edges

Min-Cut-Max-Flow Theorem

- Flow of cut <= capacity of cut
- Flow of cut == Flow of network
- Hence, Capacity of min-cut = max-flow of network
- Ford-Fulkerson terminates when there is cut that meets requirements → Flow of each outgoing edge == capacity of edge, flow on each incoming edge to cut is 0 (we call this min-cut)
- Easy way to do this is. Find all outgoing edge from s, put in one section. Others all put along with t.
- Max-Flow is something that most likely won't come out in exam, and the algos aren't covered in 2004.

Bipartite Graph

Nothing much tbh :P



Why are these important?

- **Network Flow** → Very practical application in a lot of real-life scenarios (e.g. Piping, "Network" Flow, etc.)
- **Bipartite Graph** → Same as Network Flow

What's Next?

- **TBC**

Min-Cut

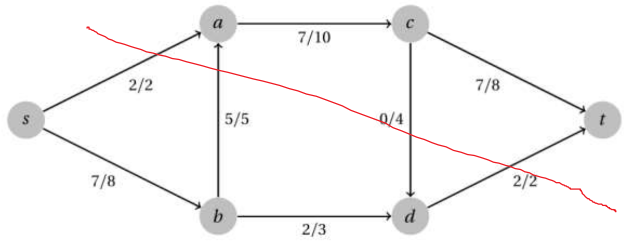
Wednesday, 1 June, 2022 21:15

Network Flow

Question 22

1.5 Marks

Which edges comprise the **minimum cut** in this network?



Select one or more:

1. s->a
2. s->b
3. b->a
4. a->c
5. b->d
6. c->d 5
7. c->t
8. d->t

Find the cut that set {s} means s can reach to the vertex and the other set {t} is unreachable from {s}.

In this case {s,b,d} and {t,a,c}