Assignment 3 Will be returned from the Department Office.

Check the class homepage

Test 3: Next week

- Need 40% to pass the course
- Materials:
  - 25% sorting + divide-and-conquer
  - 25% greedy + dynamic programming
  - 50% graph search + flow network.

Additional notes from U of T on the web

Flow Network: ;directed graph $G = (V, E)$ where

- each edge e has a capacity $c(e) > 0$;
- the source s, sink t and *internal nodes*;
- no edge entering s, nor leaving t;
- for every internal node v, there is a path $s \to v \to t$.

Flow f

1. **Capacity condition** for edge e: $0 \le f(e) \le c(e)$
2. **Conservative condition** for internal node v:

$$\sum_{e \text{ into } v} f(e) = \sum_{e \text{ out of } v} f(e)$$

Value of f: the total flow out of the source,

$$Value(f) = \sum_{e\ out\ of\ s} f(e)$$

Lemma 1 For a cut (A, B): $Value(f) = f^{out}(A) - f^{in}(A)$

Corollary 1 $Value(f) = f^{in}(B) - f^{out}(B)$.

Corollary 2 $Value(f) = f^{in}(t)$.

Corollary 3 $Value(f) \leq c(A, B)$.

The Maximum Flow Problem

Given a flow network, find a flow of maximum value.

By Corollary 3: maximum flow value $\leq$ min capacity of any cut

Will show that they are the same

Important Facts about Ford-Fulkerson Algorithm: When all capacities are integers, produces a maximum flow, where *all flows on the edges are integers*.

Solving the Bipartite-Matching/Dancing-Party Problem

1. Construct the flow network.
2. Specify the algorithm to find a maximum flow (Ford-Fuldkerson, or others – today)
3. Construct the solution from the maximum flow.
4. **Correctness**: Prove relationship between the max flow returned by the algorithm and an (optimal) solution to the original problem.

Running time: The running time is the total running time of steps 1, 2 and 3.