April 8, 2019

Prelims 2 Review

Topics

→ Divide and Conquer

(a) Integer multiplication
(b) Fast Fourier Transform
(c) Closest pair of points in 2-D plane.

Expectation: Run times of the above algorithms

→ Not test you to run above alg. on sample inputs.

→ Design an algorithm for problem X.

- Analyzing run times: refresh recurrence solving skills.

→ Look at HW 5 problems.

* problem in prelim (if any) will be easier than HW problems.

→ Network Flow. [2 weeks of classes]
HW6, HW7.

* Important topic

Basics. : (i) Def^n of a flow network.
(ii) " " an s-t cut
(iii) Value of a flow./Def^n of a flow.
(iv) Capacity of an s-t cut.

* Ford-Fulkerson algorithm.

→ Residual graph is (w.r.t a flow f)

→ Augmenting paths.

→ Run time of F-F. pseudo poly time

$$O(\underset{\sum_{v \in V} c(s,v)}{C} \cdot \underset{\text{\# of edges}}{m})$$

* Max-Flow Min-Cut theorem.

For any flow f on a flow network

G, and any s-t cut (A, B) of

G,

(i) $v(f) \leq c(A, B)$.

(ii) Value of max. flow = Capacity of min-cut.

* Algorithm to find min-cut.

f is the max-flow on G;

$A = \{ v \in V : s \rightsquigarrow v \text{ in } G_f \}$.

$B = V \setminus A$.

<u>Lemma:</u> (A, B) is a min-cut of G.

<u>Integrality Thm:</u> If all capacities are integers, then $\exists$ an 'integral' max-flow.

<u>Edmonds Karp:</u> $O(m^2 n)$.

* Bipartite matching; Hall's condition.

(*) Important!!

Solving problems by reducing to max-flow \ min-cut.

General sheme:

Given problem X [feasibility problem or optimization problem]

'Assigning resources to jobs'.

Idea: (i) Construct a flow network G from input instance of X.

(ii) Compute max-flow on G.

(iii) Infer 'Yes / allocation'

or 'impossible'.

* Analyze run-time.

Proof of correctness:

($\Rightarrow$) Given an 'allocation' by your algorithm, show that it satisfies Problem X.

($\Leftarrow$) Given a valid solution of Problem X, show that corresponding flow that is 'valid'.

<u>Example problem</u>: Assigning TA's to OH's.

→ Given set of slots: $S = \{h_1, \ldots, h_l\}$ per week; each h_i is a 1-hour slot (distinct).

and each TA has an avail. list $T_i \subset S$.

Constraints:

(i) At most 1 TA / OH.

(ii) each TA holds at most 'a' OHs / per week.

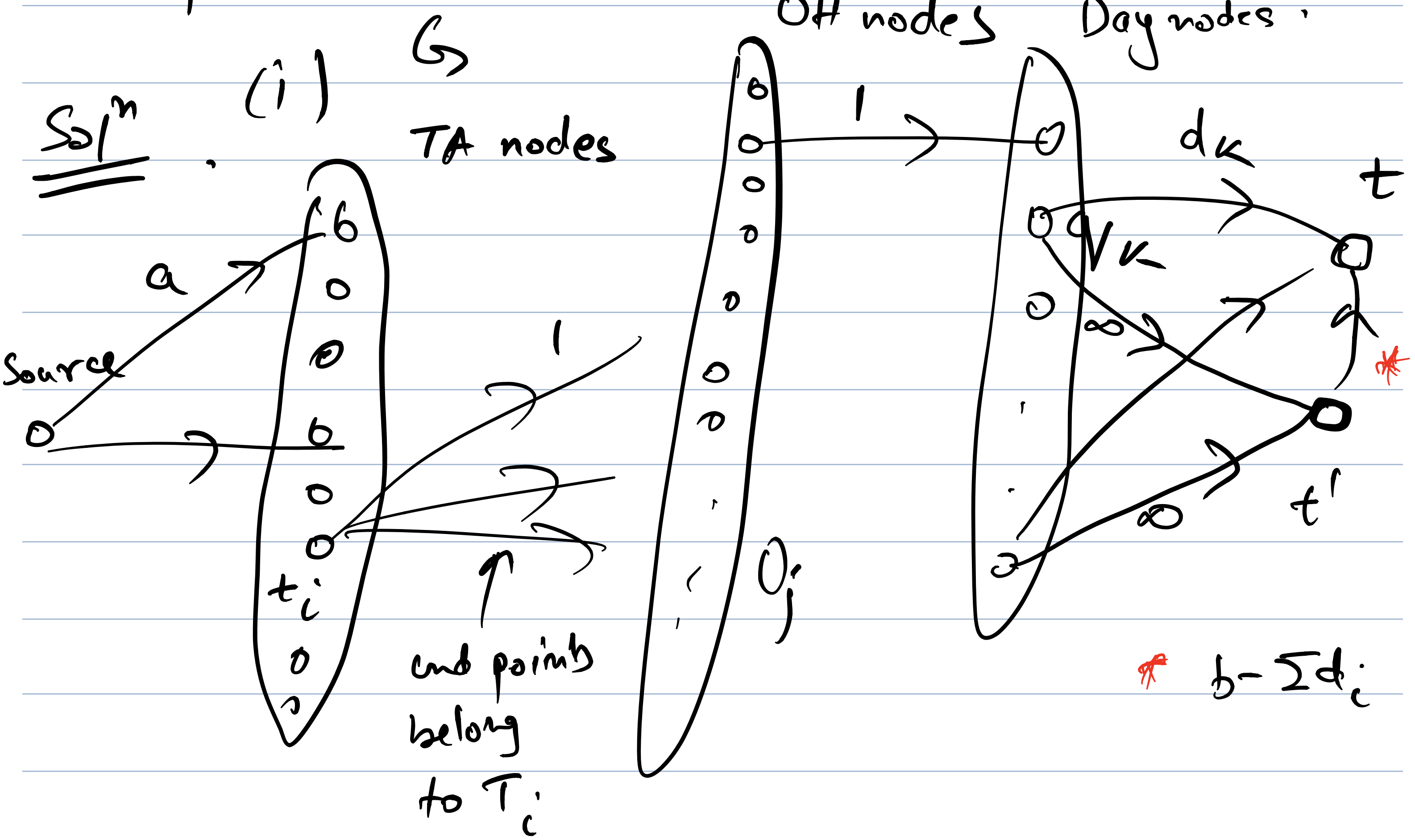
(iii) Day i requires at least d_i OHs.

(iv) Exactly b OHs / per week.

(Assume: $\sum d_i \le b$).

Output: Valid allocation or impossible.

Sol^n:



(ii) Compute f, $\boxed{\text{integral}}$ max-flow on G.

(iii) If f saturates all

incoming edges to t,

then 'Yes', allocation is

assign TA i to OH j

iff $f(t_i, o_j) = 1$.

else

'Impossible.'

Running time: $O(C \cdot m)$

$\boxed{\text{Please describe nodes and edges in your flow network.}}$

Proof of correctness.

($\Rightarrow$) suppose algorithm outputs an allocation of TAs to OHs.

Argue: * every OH is allocated to at most 1 TA.

* since Every edge to t is sat.

($\Leftarrow$) Suppose $\exists$ a feasible allocation of TA's to OHs.

Push a flow of 1 from t_i to o_j for each such allocation.

Push other flows to maintain flow conservation.

→ NP- Completeness [Important !!]

Basics : (i) Def^n of Decision problem,

(ii) Def^n of NP

Properties (iii) $\leq_p$ [poly time Karp reduction]

(iv) Def^n of NP completeness and NP-hard.

FACTS: 3SAT is NP- complete.

Feel free to use NP - completeness of any problem you have seen in class / HW / book.

Warning: Direction reduction.

$A \leq_p B$. [B is as hard as A]

Recipe to show NP- completeness of Problem X :

(i) Show $X \in NP$ [1 liner for verifi]

(ii) Pick a convenient know NP- complete / NP- hard problem A.

Prove: That $A \leq_p X$.

A : (i) 3 SAT or other similar flavour problem.

(ii) IS / Clique.

(iii) vertex cover / set cover

(iv) Hamiltonian Cycle -

(v) Subset Sum.