20 Feb 2018

Prelim 1 review

Thurs, 2/21, Statler Auditorium (Not Statler 196)

7:30 - 9:00.

closed book, closed notes.
bring a pen/pencil (+ spare)

Topics. Chapters 1, 4, 6. (Not RNA, not recurrences)
Boruvka & Floyd-Warshall in lectures, not book.
$\therefore$ not on prelim!

Algorithms we taught

Stable Matching	Gale - Shapley
Greedy	Earliest Finish Time (int'vl sched)
	Boruvka, Kruskal, Prim } MST
	Huffman (prefix coding)
Dynamic Programming	Weighted interval scheduling
	Knapsack
	Bellman-Ford
	Floyd-Warshall
	RNA secondary structure
	(* read about CYK some time!)
	(but not for Prelim 1...)

Types of questions.

- Straight "recall facts from your memory"

  E.g. Running time of Bellman-Ford on a graph with n vertices, m edges is:

  (A) $O(m+n)$ (C) $O(mn)$

  (B) $O(n \log n)$ (D) $O(n^3)$

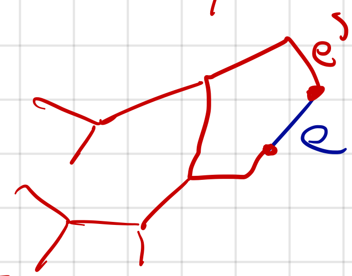
- Run a "named algorithm" on a sample input.
- We present an incorrect algorithm. You find a counterexample.
- We present a correct algorithm. You figure out its running time.
- We give you a simple fact that we want you to prove or give counterexample.

Ex 4. Prove or disprove. If G is a connected graph with at least one edge with distinct edge costs, the minimum cost edge belongs to every Min Sp. Tree.

(A) Prove (B) Disprove

Proof 1. If $e = (u,v)$ is the min cost edge, then e is the min cost edge crossing the cut from $\{v\}$ to $V - \{v\}$. By Cut Property, $e \in MST$.

Proof 2. By contradiction. Assume $e \notin T$. Let e' be any other edge in the cycle that e forms with T. Then $T - \{e'\} \cup \{e\}$ is a spanning tree and its cost is less than that of T.



Proof 3. Kruskal's Algorithm works. It always selects the min-cost edge. Hence that edge belongs to one MST. That is the only MST ~~because~~ G has distinct costs.

Uniqueness? When G has distinct edge costs, the MST is always unique.

- Design & analyze an algorithm.

Use if you want.
Prelim won't require it.

Greedy   Dyn Prog   Reduction

Bellman-Ford is a great target for reductions.

(Problem 6.? from K&T)

String of words. $w_1, w_2, \ldots, w_n$.
character counts: $c_1, c_2, \ldots, c_n$.

Print text in a format with $\leq c_{max}$ characters per line. (with 1 space btw adj words on a line).
Avoid "ragged right edge".
Line of text with c_{tot} characters is penalized by $(c_{max} - c_{tot})^2$.

Minimize sum of line penalties.
Keep words in same order, just insert line breaks.

Greedy: keep inserting line breaks at latest possible time until you run out of words.

ALGORITHMS ARE COOL.
$c_1 = 10$ $c_2 = 3$ $c_3 = 4$ $C_{max} = 14$.

ALGORITHMS ARE //
COOL

$$\begin{array}{r} 0^2 \\ +10^2 \\ \hline 100 \end{array}$$

ALGORITHMS //
ARE COOL

$$\begin{array}{r} 4^2 \\ 6^2 \\ \hline 52 \end{array}$$

:P