E0 225 - Design and Analysis of Algorithms
First Midterm Examination

21st September 2023

Total Marks: 30
Time: 80 minutes

Solve any 3 (out of the given 4) questions. All questions carry equal marks.
Please do not use pencil for answering questions.
Solutions written in pencil will not be evaluated.
Clearly mention the assumptions that you make.

Student Name: Manish Kumar

S. R. No.: 21044

Problem	1	2	3	4
Max Marks	**10**	**10**	**10**	**10**
Marks	3	2	10	-

$\frac{15}{30}$

Final answers must be contained in this exam booklet.
You may also ask for extra sheets for rough work.

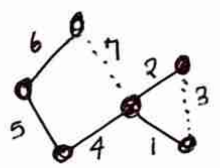
Problem 2

Minimum Spanning Tree. Let $G = (V, E)$ be a connected graph with distinct edge weights.

(i) Prove that for any cycle in G, the minimum spanning tree of G excludes the maximum-weight edge in that cycle.

(ii) Prove or disprove: The minimum spanning tree of G includes the minimum-weight edge in every cycle in G.

(*10 points*)

→ 

Dummy example

(i) → Let MST be $T \in G$

Let $e' \notin T$ ($e' \exists$ edge $\in G$)

→ $T+e'$ must contain a cycle, as 'T' is MST

→ each edge $\in G$ must be of distinct weight.

→ Let the cycle be $C \in G$,

→ Via construction of cycle 'C', $e' \in C$.

→ If we delete $e \in C$ and ~~ee~~ $e \neq e'$, then we again get a tree, say T'.

2

why?

weight $(T) \neq$ weight (T'), As MST is always unique for distinct weighted graph.

→ $\Rightarrow W(T) < W(T')$

→ This is true for any tree created via edge removal from cycle C. [Assume e' is kept in tree]

Constrain

→ None of the trees created via such edge removal technique from cycle 'C' have same weight. [Say tree T', T'', T'''...

→ Among T, T', T'', T'''..., 'T' has minimum weight [also $e' \in T$]

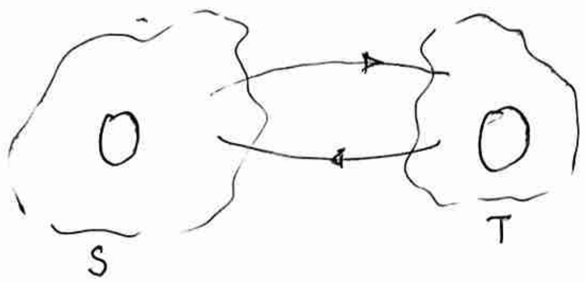
→ Constrain and unique weight of T, T', T'', ... $\Rightarrow$ MST must avoid maximum wighted edge in C

→ This is true for any cycle formed in G.

Problem 3

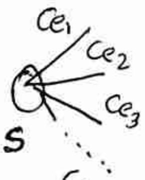
Minimum cut. Let $G = (V, E)$ be a directed graph, with source $s \in V$, sink $t \in V$, and positive integer edge capacities $\{c_e\}$. Give a polynomial-time algorithm to decide whether G has a *unique* minimum s-t cut (i.e., an s-t cut of capacity strictly less than all other s-t cuts). (*Hint: Perturb/modify the capacity of some edge(s).*)

(*10 points*)



→ Unique min-cut should have symmetry with respect to perturbation to capacity modification at edges of source and sink.

Identification Algorithm :



(I) perturb c_{e_1} by 1 unit and measure min cut location, say $m\text{-}cut_1$.

(II) Do it for c_{e_2} ... upto c_{e_n}

(III) check if $m\text{-}cut_1 = m\text{-}cut_i \ \forall i \in N$
↳ if all are same or not

(IV) If they are same, Declare unique min-cut exist in the flow-network.

10

Algorithm is poly-time as $m\text{-}cut_i$ check is polytime.
~~There a~~ ~~Atmost~~ order 'n' check required.

E0 225 - Design and Analysis of Algorithms
Second Midterm Examination

26th October 2023

Total Marks: 30
Time: 80 minutes
Please do not use pencil for answering questions.
Solutions written in pencil will not be evaluated.
Clearly mention the assumptions that you make.
Attempt all the three questions.

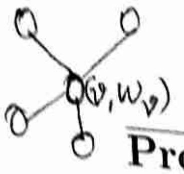
Student Name: Manish Kumar

S. R. No.: 21044

Problem	1	2	3
Max Marks	**10**	**10**	**10**
Marks	9	5	3

17

Final answers must be contained in this exam booklet.
You may also ask for extra sheets for rough work.



Problem 1

In the minimum weight vertex cover problem, the input is an undirected graph $G = (V, E)$ and weights $w_v > 0$ for all vertices $v \in V$. The goal is to select a subset of vertices $S \subseteq V$ of minimum weight such that for each edge (u, v) either $u \in S$ or $v \in S$. The following LP was used in the class to design an approximation algorithm for the problem.

$$(LP) \quad \min \sum_{v \in V} w_v x_v$$
$$\text{s.t.} \quad x_u + x_v \geq 1, \quad \forall (u, v) \in E \quad \longrightarrow \text{Constrain-1}$$
$$x_v \in [0, 1] \quad \longrightarrow \text{Constrain-2}$$

✓

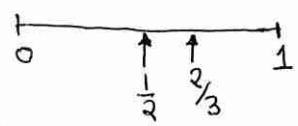
In this problem, we will explore other possible ways to round the fractional solution $\{x_v\}_{v \in V}$ returned by the LP.

1. For each vertex $v \in V$, let $x'_v = 0$ if $x_v < 2/3$, and $x'_v = 1$ if $x_v \geq 2/3$. Report all the $v \in V$ whose $x'_v = 1$. Will it be a feasible solution? If yes, then what is the approximation factor? If no, then justify.
2. For each vertex $v \in V$, let $x'_v = 0$ if $x_v < 1/3$, and $x'_v = 1$ if $x_v \geq 1/3$. Report all the $v \in V$ whose $x'_v = 1$. Will it be a feasible solution? If yes, then what is the approximation factor? If no, then justify.

(10 points)

Part-I:

$$x'_v = \begin{cases} 0 \; ; & x_v < 2/3 \\ 1 \; ; & x_v \geq 2/3 \end{cases}$$



$\frac{1}{2} = 0.5$

$\frac{2}{3} = 0.666\ldots$

<u>NOT</u> a feasible Solution

Reason: Counter-example exists

take $x_v = 0.55$
$x_u = 0.45$

It satisfy $x_v + x_u \geq 1$ $[0.55 + 0.45 = 1]$
$x_v, x_u \in [0, 1]$

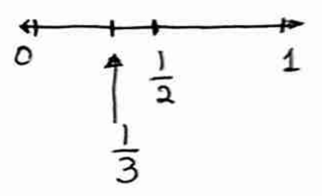
But $x'_v = 0$ and $x'_u = 0$

$\Rightarrow x'_v + x'_u = 0$

Or, it is possible that algorithm output neither vertex 'u' or 'v' associated with edge $e = (u, v)$
This voilates ~~vent~~ min-weight 'vertex cover' definition/criteria.

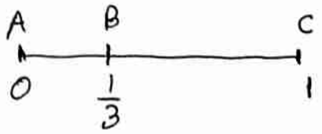
Part 2:

$$x_v' = \begin{cases} 0 ; & x_v < \frac{1}{3} \\ 1 ; & x_v \geq 1/3 \end{cases}$$



YES, a feasible solution.

$\boxed{\text{Reason : Proof exists}}$



~~Lets pick~~ Logic : ~~constrain-I implies there is two~~

→ Let assume (without loss of generality)

x_v lies in region AB. $[0 \leq x_v \leq \frac{1}{3}]$

$\Rightarrow x_u$ must be in region BC. $[\frac{1}{3} \leq x_u \leq 1]$

This is due to Constrain-I. $[x_v + x_u \geq 1]$

→ Thus

x_v' always output atleast 1 for all edge $e \in (u,v)$

a

→ Constrain-II is trivially satisfied due to choice of x_v, x_u we are taking into ~~account~~ account.

Note: ~~If x_v lies in rego~~ If we take x_v in region BC. There is no need to bother about x_u. As x_v' will anyway output atleast 1. Thus Constrain-1 satisfied.

Approximation factor : It remain same. ~~as like~~

$OPT_{LP} \leq 3 \cdot OPT_{(actual)}$ → Reason

~~Reason~~ :

~~$x_v' + x_u' \leq 2(x_v + x_u)$~~

example.

Because, ~~$x_v' + x_u' \leq 2$~~ ~~$[x_v = 0.5, x_u = 0.5]$~~

~~$x_v' \geq 3x_v$~~

~~$x_u' \geq 3x_u$~~

~~$x_v' + x_u' \geq 3(x_v + 3x_u)$~~

~~$x_u' \geq 3x_u$~~

$x_v' \leq 3x_v$

$x_u' \leq 3x_u$

$x_v' + x_u' \leq 3(x_v + x_u)$

Final step missing

$OPT_{LP} \leq 3 OPT$

Problem 2

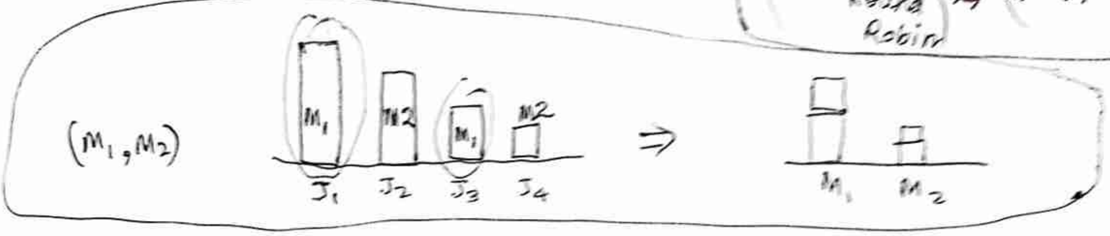
As discussed in the class, the goal in the load balancing problem is to minimize the makespan, i.e., minimize the maximum time taken by any machine to finish the tasks assigned to it. Prove that the approximation factor of the following round-robin algorithm is two.

Let $J_1, J_2, \ldots, J_n$ be n jobs with decreasing processing times $t_1 \geq t_2 \geq \ldots t_{n-1} \geq t_n$. Let there be m machines $M_1, M_2, \ldots, M_m$. For simplicity, let $n = \beta \times m$, where $\beta \geq 1$ is an integer. Consider an approximation algorithm which allocates jobs in a round-robin fashion to the machines: For all $1 \leq i \leq m$, machine i is assigned n/m jobs $J_i, J_{i+m}, J_{i+2m}, J_{i+3m}, \ldots, J_{i+(\beta-1)m}$.

For example, if there are nine jobs and three machines, then machine M_1 is assigned J_1, J_4 and J_7, machine M_2 is assigned J_2, J_5 and J_8, and machine M_3 is assigned J_3, J_6 and J_9.

(10 points)

→ $\beta = \frac{n}{m}$; To show $\left(OPT_{\text{Round Robin}} \leq 2 \cdot OPT_{\text{actual}} \right)$



Dummy example.

→ Since, Jobs are sorted $[t_1 \geq t_2 \geq \cdots]$, The bussiest machine is M_1. [Note: $t_1 \geq t_j$ and $t_{1+m} \geq t_{j+m}$] $(\forall m, j)$

Thus M_1 will determine makespan.

→ Let T^* is optimal solution.

✩ we analyse the case when we place a job J_{1+j} on M_1, where j is as per round-robin algo.

Observation-I: By ~~Construction/ &~~ ordering of time,

$$t_{1+j} \leq T^* \quad \rightarrow (1)$$

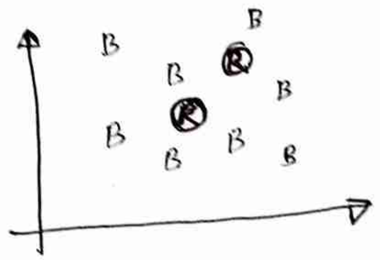
[As, t_1 is already placed on M_1, and $t_{1+j} \leq t_1$] $\forall j$.

Problem 3

(*Skyline points*) Let B be a set of n blue points in 2D and let R be a set of m red points in 2D. Also, $n \gg m$. Design an $O(n \log m)$ time algorithm to determine *all* the blue points which are dominated by at least one red point. A red point $r = (r_1, r_2)$ is said to dominate a blue point $b = (b_1, b_2)$ iff $r_1 > b_1$ and $r_2 > b_2$.

(10 points)

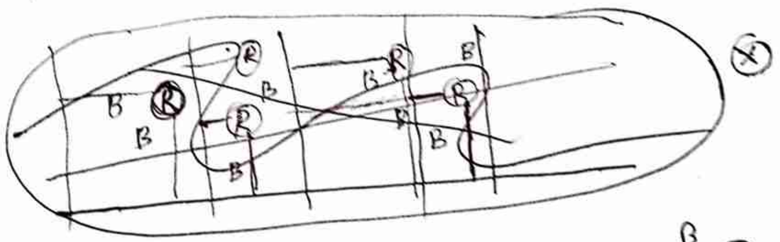
→ # bule point >> # red point

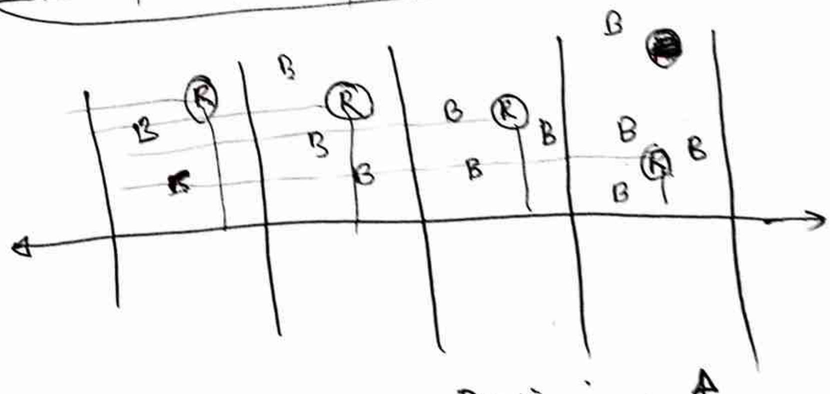


Ⓡ = Red Point

chan algo: $O(n \log \hat{k})$

We will use chan algorithm type strategy. Number of red points are analogous to $\hat{k}$ partition for blue point.





Pastioning ↑
(By chan method)

Manish Kumar (21044)

E0 224: Computational Complexity Theory
Indian Institute of Science
Final exam 2023

Duration: 3 hrs 15 mins **Total marks: 60**

Note: You may assume any result covered in the lectures or the assignments.

1. **(15 marks)**

   (a) **(5 marks)** Consider the following problem: Given two positive integers q and r, check if q has a factor smaller than r. Show that if the problem is NP-hard, then PH collapses.

   (b) **(5 marks)** Prove that PSPACE $\neq$ (logspace uniform)NC^1. (Hint: Recall the hierarchy theorems.)

   (c) **(5 marks)** Prove that if $\oplus\mathsf{P} \subseteq \mathsf{BPP}$, then $\mathsf{PH} = \Sigma_2 \cap \Pi_2$. (Hint: Toda's theorem)

2. **(10 marks)** Show that the following problem is in NL: Given an undirected graph G, vertices s and t, and a positive integer k, check if the shortest path from s to t in G has length k. (Hint: NL =?)

3. **(10 marks)** Let $k \leq n$. Prove that the following family $\mathcal{H}_{n,k}$ is a collection of pairwise independent hash functions from $\{0,1\}^n$ to $\{0,1\}^k$. Identify $\{0,1\}$ with the field $\mathbb{F}_2$. For every $k \times n$ matrix A with entries in $\mathbb{F}_2$, and $\mathbf{b} \in \mathbb{F}_2^k$, $\mathcal{H}_{n,k}$ contains the function $h_{A,\mathbf{b}} : \mathbb{F}_2^n \to \mathbb{F}_2^k$ defined as $h_{A,\mathbf{b}}(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$ for $\mathbf{x} \in \mathbb{F}_2^n$.

4. **(12 points)** Let $A = (a_{ij})_{i,j} \in \mathbb{F}^{n \times n}$ be an $n \times n$ matrix over a field $\mathbb{F}$. The *permanent* of A is defined as

$$\mathrm{perm}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i,\sigma(i)},$$

Schwartz Zippel

where S_n is the group of all permutations of $1, \ldots, n$.

Show that there is a randomized algorithm that, given an oracle that can compute the permanent correctly on $1 - \frac{1}{3n}$ fraction of the inputs in $\mathbb{F}^{n \times n}$, can compute the permanent on *any* input correctly with high probability using poly(n) operations over $\mathbb{F}$. Assume that $\mathbb{F}$ has more than $3n$ elements.

(Hint: Let A be an input matrix. Pick a random matrix $R \in_r \mathbb{F}^{n \times n}$ and consider the matrix $A + xR$, where x is a formal variable.)

5. **(13 marks)** Define the Majority function $\mathrm{Maj} : \{0,1\}^n \to \{0,1\}$ as follows:

$$\begin{aligned} \mathrm{Maj}(x_1, x_2, \ldots, x_n) &= 1 \quad \text{if more than } \frac{n}{2} \text{ of the arguments are 1,} \\ &= 0 \quad \text{otherwise.} \end{aligned}$$

Show that any depth-d AC^0 circuit computing $\mathrm{Maj}(x_1, x_2, \ldots, x_n)$ has size $2^{n^{\Omega(\frac{1}{d})}}$.

(Hint: Show that if Majority has a small depth-d circuit, then PARITY has a small depth-d circuit. How? Use the circuit for Majority to create a circuit $C_k(x_1, \ldots, x_m)$ that outputs 1 if and only if $\sum_{i \in [m]} x_i = k$.)