B-TECH 4TH SEMESTER MID-TERM EXAM

April 2021

Subject: Analysis and Design of Algorithms [CS2201]

Date: 15/04/2020

Name: Abhiroop Mukherjee

Enrolment Number: 510519109

G-Suite ID: 510519109.abhirup@students.iiests.ac.in

No. of Sheets Uploaded: 9

→ Roll no: 510519109

∴ $x = 1$
$y = 0$
$z = 9$

Q1) a) ~~Median~~ Median of Median

Time complexity: O(n) → merit

→ Does insertion sort, if division group, big, it goes toward O(n²) → demerit

Randomized Pivot sel

Time Complexity: Average: O(n)
Worst: O(n²) → demerit

→ No problem due to sorting → merit

Q1) b) given numbers.

23, 71, 42, 37, 109,
59, 67, 35, 29, 10,
17, 26, 34, 82, 91,
12, 43, 61, 28, 1,
25, 40, 51, 93, 90

[assuming concatenation m42]

→ as we are taking 5 elements per group, we first sort these groups and take median of the ~~5~~ five groups to make a new set.

23, 37, 42, 71, 109
10, 29, 35, 59, 67,
17, 26, 34, 82, 91,
1, 12, 28 43, 61,
25, 40, 51 90, 93

→ so new group: 42, 35, 34, 28, 51

→ now we sort the new group and find ~~it~~ it's median

28, 34, 35, 42, 51

∴ end result of MoM: 35

→ Now finding actual median by sorting 25 numbers...

1, 10, 12, 17, 23, 25, 26, 28, 29, 34, 35, 37, 40, 42, 43

51, 59, 61, 67, 71, 82, 90, 91, 93, 109

∴ actual median: 40 → 13th pos.

MoM output: 35 → 11th pos.

∴ error: 2 positions

Q2) a) given following things

i) Tasks that will be done in unit time

ii) Deadline for each tasks

iii) Penalty for ~~doing~~ not doing task (non-ve)

and we need to minimize penalty.

→ We see that ~~these~~ doing and not doing works creates two sets

i) Early Tasks: Tasks done before deadline

ii) Late Task: Task which will have penalty taken

→ we see that minimizing late task penalty means maximising Early Task Penalty

→ so we can try to use greedy method on Tasks to create max penalty in Early Task set

→ Let N_t be no. of tasks in task set A whose deadline is t or earlier.

→ we see that if $N_t(A) > t$, for some t, set of early tasks is not feasible.

Eg 3 tasks.

deadline = 2, 1, 2 = (A)

$N_1(A) = 1$

$N_2(A) = 3$ (>2) → not feasible

→ So we can use this property to check for compatibility.

→ Now we try to link this problem to Matroids, as we know matroids can have greedy property

→ Let ~~task se~~ early task set E be independent, if

$\cancel{N_t}$ $N_i(E) > t_i$ ~~∀ i in deadline range (1, max(dead~~

for all i in range 1 to max (deadline)

I) Every Subset of E aslo follows independance, as they also have $N_i(E') > t_i$ $\forall i$, $E' \subset E$

II) <u>Hereditary</u>

→ if B is independant and some $A \subseteq B$, then A will also ~~follow~~ have no deadline property

III) Exchange

→ suppose A and B independant
with $|B| > |A|$

→ Let k be largest t such that
$$N_t(B) \leq N_t(A)$$

→ we must have some $k < n$, where
$$N_j(B) > N_j(A) \quad \forall j \text{ in } k+1 \leq j \leq n$$

i.e B contains more tasks with deadline k+1 than A does.

→ Let $n \in B - A$, be a task with deadline k+1

Let $A' = A \cup \{n\}$

→ For range $1 \leq t \leq t$, $N_t(A') = N_t(A)$, hence it follows deadline propertr.

→ For range $k < t \leq n$, $N_t(A') \leq N_t(B)$ as B follows deadline propert, A' will also follow deadline property

→ Hence A' is independant and Exchange Property is Proved

→ Hence as we have linked Task Scheduling Problem to Matroid, which we know can be used for greedy Algorithm, we can use greedy Algorithm on this problem as well

b) Givenn $[x=1, y=0, z=9]$ [concatenation]

Tasks	T1	T2	T3	~~2~~ T4	T5	T6
Deadline	1	3	3	2	1	4
Penalty	121	29	50	60	25	35

→ sorting Based on Penalty [Descending]

Tasks	T1	T4	T3	T6	T2	T5
Deadline	1	2	3	4	3	1
Penalty	121	60	50	35	29	25

now we do the following.

Task Set	N_1 (≤ 1)	N_2 (≤ 2)	N_3 (≤ 3)	N_4 (≤ 4)	Remarks
T1	1	1	1	1	Independant
T1, T4	1	2	2	2	Independant
T1, T4, T3	1	2	3	3	Independant
→ T1, T4, T3, T6	1	2	3	4	Independant
T1, T4, T3, T6, T2	1	2	4	5	Not Independant
T1, T4, T3, T6, T5	2	3	4	5	Not Independant

∴ Early Task ⇒ T_1, T4, T3, T6

Late Task ⇒ T2, T5 ⇒ Penalty = 54

Schedule [sort Early Task Based on Deadline Ascending]

⇒ T1, T4, T3, T6

3) a) Eg

$$A_1 \times A_2 \times A_3 \times A_4 \times A_5$$

2×3 3×4 4×2 2×6 6×2

→ when we ~~do~~ try to test all the possibility, we do a lot of repetations.

→ ~~Eg~~ suppose we do following split

$$(A_1 A_2 A_3) \times (A_4 A_5)$$

→ we see that for finding bigger cases, we need to calculate smaller one

$\therefore C_5 = C_3 C_2$, wher $C \to$ no. of ways

⇒ to find all the ways, we try every possible split

$$\therefore C_k = \sum_{i=0}^{n} C_i C_{k-i}$$

which give $C_i = \frac{1}{i+1} \, {}^{2i}C_i$

→ which using striling's approximation, we ~~$i! \approx \sqrt{2}$~~

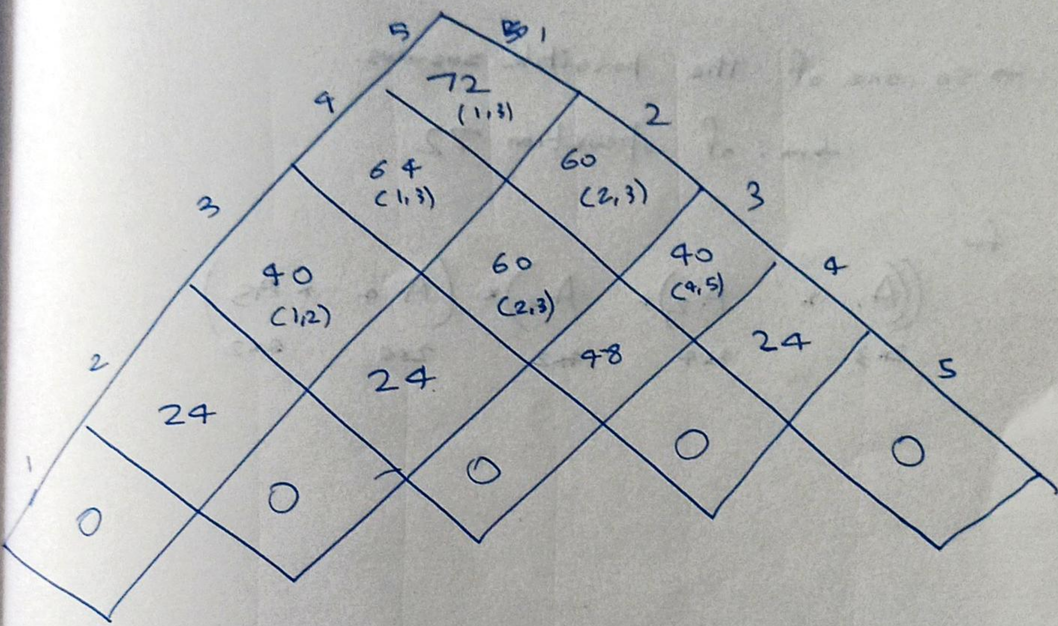
see that ~~Eg~~ calulating C_i is exponential

2) b) Given matrices [x = 1 , y = 0 , z = 4]

$A_1 \times A_2 \times A_3 \times A_4 \times A_5$

2×3   3×4   4×2   2×6   6×2

→ we ~~make~~ do the following using m-table.



i) for $A_1 \times A_3$ → any work as values same

24 + 0 + 16

ii) for $A_2 \times A_4$

24 + 0 + 3×2×6 vs 48 + 0 + 3×4×6

60 ✓ vs 120

iii) for A_3, A_5

48 + 0 + 2×6×2 vs 24 + 0 + 4×2×2

72 vs 40 ✓

iv) for $A_1 A_4$

40 + 0 + 2×2×6 vs 24 + 48 + 2×4×6 vs 0 + 60 + 2×3×6

64 ✓ vs 120 vs 96

v) for , $A_2 A_4$

60 + 0 + 3×6×2 vs 24 + 24 + 3×2×2 vs 0 + 40 + 3×4×2

96 vs 60 ✓ vs 64

~~vi) for $A_1 A_5$~~

vi) for A_1 , A_5

$< 40 + 2\times6\times2$ vs $40+24+2\times2\times2$ vs $24+40+2\times4\times2$

vs $0+60+2\times3\times2$

88 vs 72 ✓ vs 80 vs 72 ✓

→ so one of the possible answers.

→ no. of operation 72

for

$$((A_1 \times A_2) \times A_3) \times (A_4 \times A_5)$$

2×3 3×4 4×2 2×6 6×2