MIDTERM SOLUTIONS

I. a/ Greedy - MST

b/ NO, non-constant work might be required to fill in each matrix entry.

c/ $\frac{1}{n}$, $\log n$, $\sqrt{n}$, n, n^3+n^2, $n^3 \log n$, n^{100000}

2^n, $\left(\frac{5}{2}\right)^n$, n^n

d/ $a=7$, $b=2$, $d=2$

$a > b^d \Rightarrow \Theta(n^{\log_2 7})$

e/ $a=3$, $b=9$, $d=1/2$

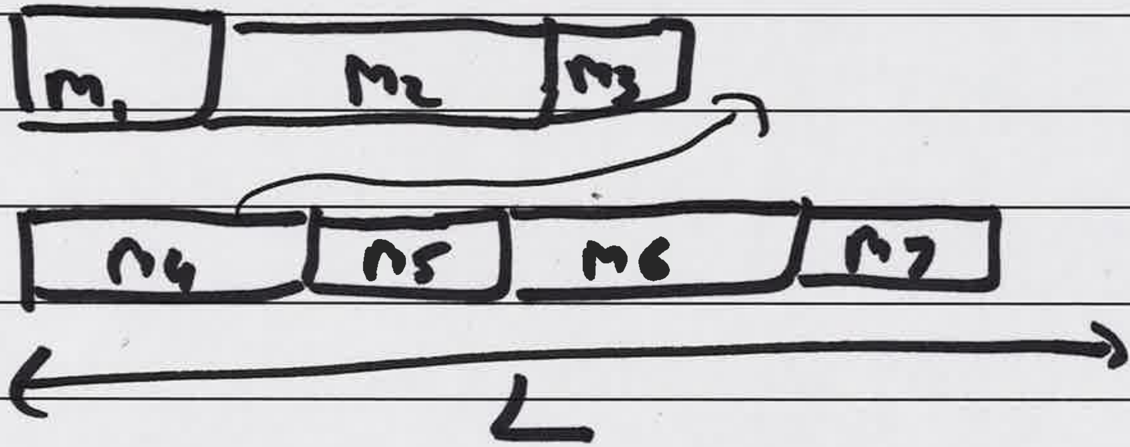
$a=b^d \Rightarrow \Theta(n^{1/2} \log n) = \Theta(\sqrt{n} \log n)$

[redacted]

2 Q2a

II. TA's computing lab

a/



<u>Greedy Algo</u> : stores as many machines as possible on first rack, starting with m_1, finishing with m_{k_1} (included)
Start again on 2nd rack with

m_{k_1+1} .....

[redacted]

Optimality : - consider any instance

- compare greedy sol. on this instance, G to an opt. sol. O.

Let i be the rank of the first rack for which the two solutions differ.

Rack i:

$G \mid m_k \quad m_{k+1} \quad \dots \quad m_p \quad \langle m_{p+1} \dots \rangle$

$O \mid m_k \quad m_{k+1} \quad \quad m_p$

O' $\quad \langle m_{p+1} \dots \rangle$

G contains strictly more machines on rack i than O.

$\Rightarrow$ we build O' as follows:

- First i racks: same machines as in G.
- Remaining Racks: same as in O

[redacted]

(4) Q2a

- except we discard machines that are on rack i.

O' : same nb of racks as O $\Rightarrow$ optimal

First $i+1$ racks identical in O' and G.

$\Rightarrow$ iterate the process

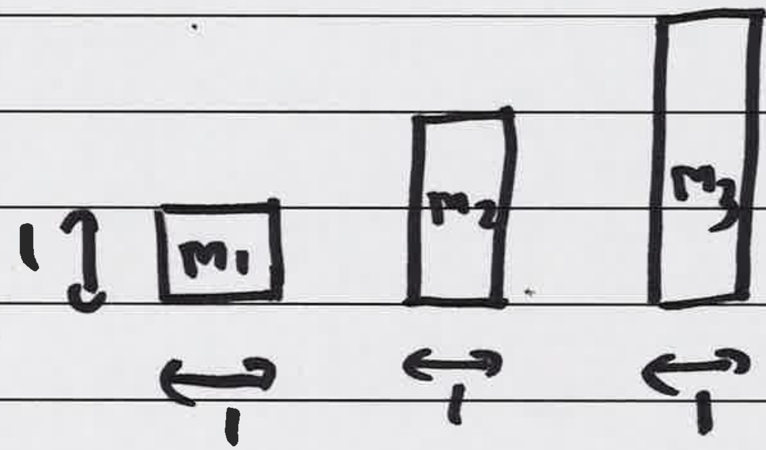
At most n racks in an opt. sol.

$\rightarrow$ G is an opt. sol.

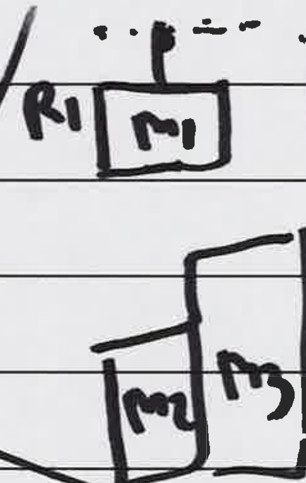
[redacted]

5 Q2b

2.b/

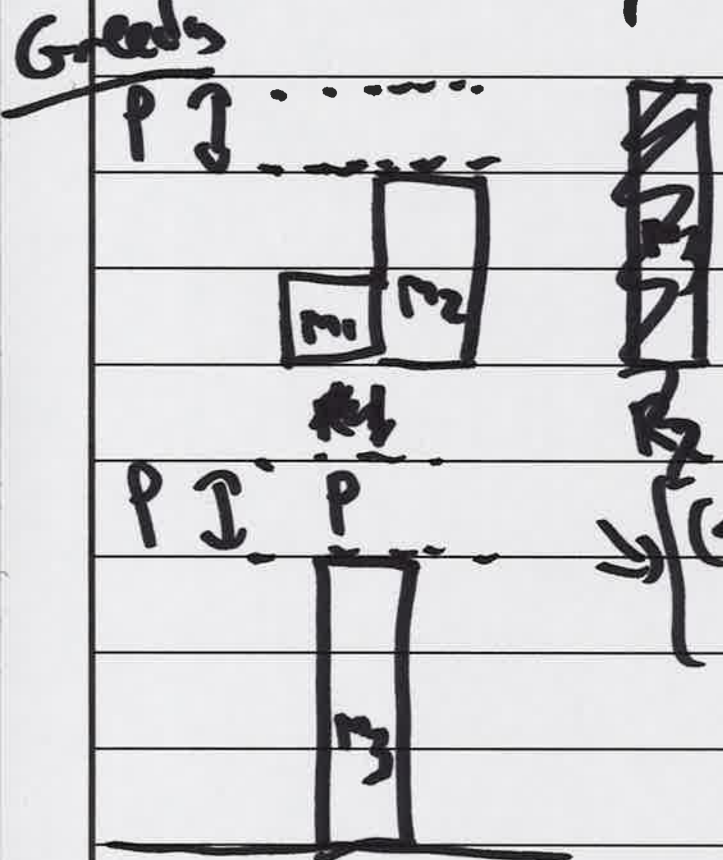


$L = 2$



$$\left((1+1) + (3+1)\right) \times 2 = 12$$

Greedy



$$\left[(2+1) \leftarrow h(\text{rack}1)+p \quad + (3+1)\right] \leftarrow h(\text{rack}2)+p$$

$\times 2$

$\uparrow$

L

$= 14$

[redacted]

6 Q2 c

II.c $S(i)$ = opt. storage cost of machines $m_i, m_{i+1} \dots m_n$

(i) Opt. substructure.

$m_1 \dots m_n$: opt. sol storing all machines

m_k is the last machine of first rack.

$m_{k+1} \dots m_n$?

$\hookrightarrow$ opt. sol. top pb $S(k+1)$

Cost. of whole sol ?

- First Rack storage:

$$\left(\text{Max height of a machine on this rack} + p\right) \times L$$

- $+ S(k+1)$

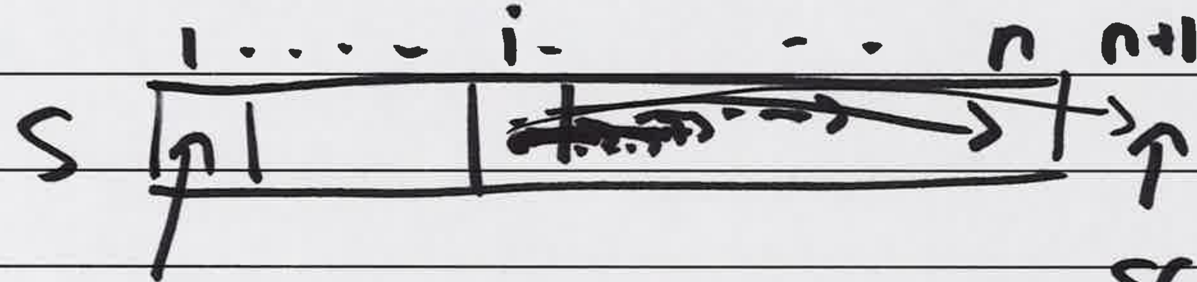
[redacted]

8 Q2c

b/ Recurrence

$m_i \cdots m_k \mid m_n$

$$S(i) = \min_{\left\{ \substack{i \le k \le n \\ \sum_{j=i}^{k} w_j \le L} \right\}} \left\{ \underbrace{(\mathrm{Max}(h_i, \dots, h_k) + p) \times L}_{\text{storage cost of current rack}} + \underbrace{S(k+1)}_{\text{cost of remaining of the sol.}} \right\}$$



The sol.

$S(n+1) = 0$ <base case>

[redacted]

c/ Complexity.

Space : $O(n)$

Time : $O(n^2)$

d/ Algorithm.

- Top-down : implement the recurrence
  - don't forget the memo
  - be careful with the max.

- Bottom-up. $LM[i]$: index of the last machine on the rack starting with m_i

[redacted]

9 Q2 c

- $S(n+1) = 0$ (base case
- For $i := n$ downto 1 {

  rackW ← w_i
  rackH ← h_i
  rackS ← $(h_i + P) \times L$
  LM[i] ← i ;

  $m_i \ldots m_n$
  only m_i on my rack

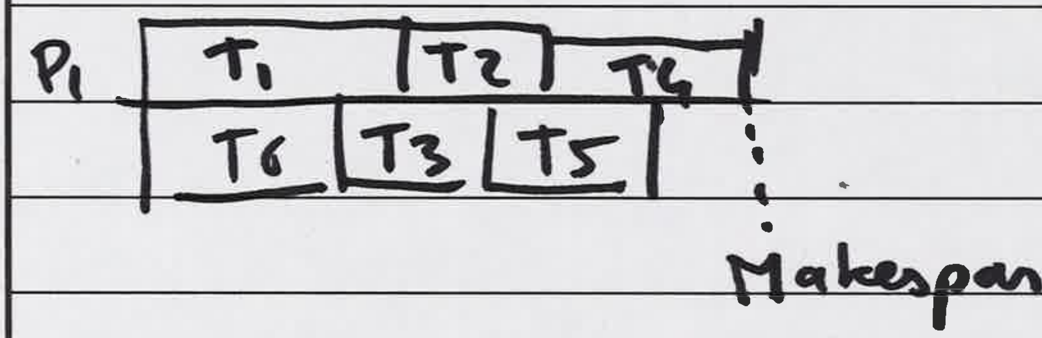
  For $j := i+1$ to n {
    rackW ← rackW + w_j
    if $h_j >$ rackH then { rackH ← h_j
      rackS ← $(h_j + P) \times L$

    if rackW $\leq L$ and
      rackS + $S(j+1) \leq S(i)$
        then $S(i)$ ← rackS + $S(j+1)$
  } LM[i] ← j

[redacted]

10 Q3

III. a/ Min. Max of 2 exec. time



IIIa. opt. pb

b. Given k (and other parameters of the pb), find a schedule of makespan not greater than k.

$\sum w_i = S$

[redacted]

11 Q3

III b. $C(i, t)$ is true if we can execute $T_1 \dots T_i$ on proc. 1 in a time less than t, false otherwise Assuming proc. 1 gets more tasks.

- $C(i, t) = C(i-1, t)$ or $C(i-1, t-w_i)$
sol. $C(n, k)$
base case

$$C(0, t) = \text{true iff}$$
$$0 \leq t \leq k - S/2$$

- Complexity $O(nk)$.

III c. Pb is in NP
We do not know whether $\in P$.

[redacted]

12 Q4

IV. a/ ii & iv are correct

[n, $m = n^2$, $\log(w(e))$]

i has k / iii has $\max(w(e))$

b/ i & ii

c/ See class notes

[redacted]