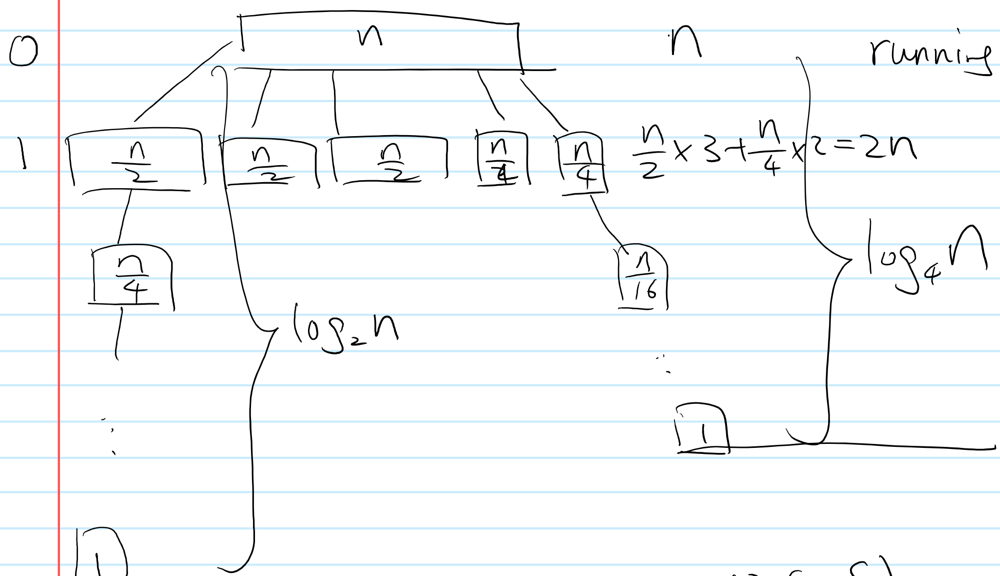
# Midterm 1 Review

Wednesday, February 12, 2020 2:57 PM

recursion (a) $T(n) = \underline{3T(\frac{n}{2})} + \underline{2T(\frac{n}{4})} + n$




running time $= \sum_{i=0}^{\#layers} \boxed{\text{merge cost at layer } i}$

$= \sum_{i=0}^{\#layers} 2^i n$

$\Downarrow$

$n^{1.5} \le T(n) \le n^2$

$T(n) = \Theta(n^c)$

$\underline{n^c} = \underline{3\left(\frac{n}{2}\right)^c} + \underline{2\left(\frac{n}{4}\right)^c} + n$

$1 = 3\left(\frac{1}{2}\right)^c + 2\left(\frac{1}{4}\right)^c$

solve for c to get the correct running time.

---

(b) T(n) be running time on an n×n grid

$T(n) = 4T(\frac{n}{2}) + \Theta(n^2)$

by Master's theorem $\log_2 4 = 2$

$T(n) = \Theta(n^2 \log n)$

---

counter-example

idea: 1 item has best ratio, but not good for fitting

10 $\left\{ \begin{array}{l} W_1 = 6 \\ \boxed{\begin{array}{l} W_2 = 5 \\ W_3 = 5 \end{array}} \end{array} \right.$ $\quad \underline{V_1 > 6 \quad V_1 < 10} \quad V_1 = 8$

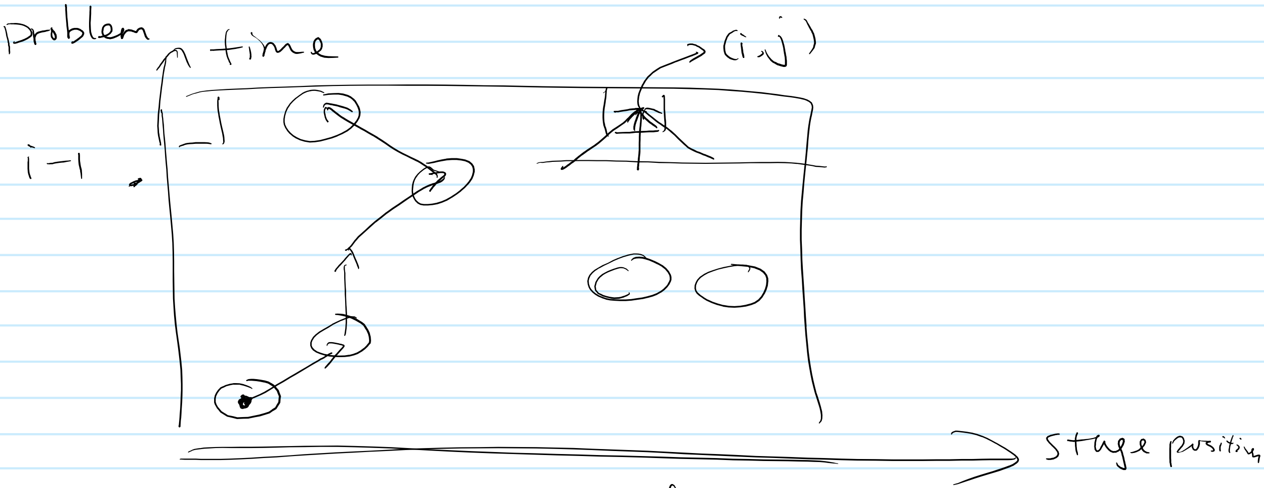
$V_2 = 5$

$V_3 = 5$

$$10 \quad \begin{cases} W_1 = 6 & V_1 > 6 \quad V_1 < 10 \quad V_1 = 8 \\ \boxed{\begin{matrix} W_2 = 5 \\ W_3 = 5 \end{matrix}} & \begin{matrix} V_2 = 5 \\ V_3 = 5 \end{matrix} \end{cases}$$

---

DP problem




let $f[i,j]$ be the max number of pancakes can get if at time $i$, the character is at position $j$.

$$f[i,j] = \max \begin{cases} f[i-1, j-1] & j > 1 \\ f[i-1, j] \\ f[i-1, j+1] & j < n \end{cases} + a[i,j]$$

$a[i,j] = 1$ if and only if there is a pancake at location $j$, time $i$

$$f[0,1] = 0 \qquad f[0,2] = f[0,3] = \ldots = -\infty$$

---

Greedy algorithm

for each group, select the smallest room that fits this group.

Proof of correctness: assume towards contradiction that there is a solution that schedules more meetings than ALG.

without loss of generality, assume algorithm considers the groups in the order $1, 2, \dots, n$.

Let $r_i$ be the room assigned to group $i$ ($r_i = -1$ if no room was assigned)

Let $p_i$ be the room assigned to group $i$ in OPT.

let $q$ be the first index where $r_i \neq p_i$

if $r_i = -1$ (ALG did not assign a room), it means

1. that there are no room that is large enough after assigning $r_1, r_2, \dots, r_{i-1}$, so OPT also shouldn't be able to assign a room.

if $r_i \neq -1$, and $p_i \neq -1$, (both ALG and OPT assigned a room)

2. by design of ALG, room $r_i$ has capacity no larger than room $p_i$

   room $r_i$ in OPT is either not used, or assigned to a group <u>later</u> than $i$ (because $r_j = p_j$ for $j < i$), call it $i'$

   in both cases we can use room $p_i$ for group $i'$ and use room $r_i$ for group $i$.

if $r_i \neq -1$ and $p_i = -1$ (ALG assigned a room but OPT did not assign a room)

3. again, room $r_i$ in OPT is either not used, or assigned to $i' > i$. We can just assign room $r_i$ to group $i$ instead

In both case 2 and case 3 we construct an alternative solution that is no worse than OPT, but agrees with ALG on at least 1 more decision. Repeating this, we eventually show ALG = OPT, this contradicts with the assumption that OPT was better.