Bus loading

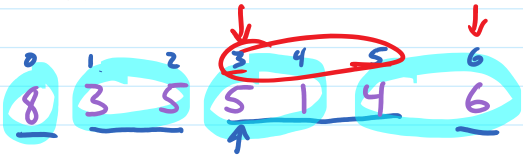
```
def load_buses(A, c):
    i ← 1
    curr ← A[0]
    B ← [0]
    m ← 1
    while i < len(A):
        if A[i] + curr > c:
            B.append(i)
            m ← m + 1
            curr ← 0
        curr ← curr + A[i]
        i ← i + 1
    return B
```

PRE: $c > 0$
elts in A are positive and $\le c$
$\text{len}(A) > 0$

POST: B is increasing with $B[0] = 0$
all values in B are legal indices into A
and no bus is overloaded
and len(B) is minimized

INV: a) B is in increasing order
b) $B[0] = 0$
c) $0 \le B[j] < \text{len}(A)$ for all j, $0 \le j < \text{len}(B)$
d) $B[m-1] < \underset{=}{i} \le \text{len}(A)$
e) all buses validly loaded for $0 \le j < \text{len}(B) - 1$ $\displaystyle\sum_{k=B[j]}^{B[j+1]-1} A[k] \le c$
f) current bus is validly loaded
$$\text{curr} = \sum_{k=B[m-1]}^{i-1} A[k] \le c$$
g) buses were loaded greedily for $0 \le j < \text{len}(B)-1$ $\displaystyle\sum_{k=B[j]}^{B[j+1]} A[k] > C$

→ h) For any optimal Θ $B[j] \ge \Theta[j]$ for $0 \le j \underset{=}{<} m$
and $\text{len}(\Theta) \ge m$



$c = 10$

$B = \; 0 \;\; 1 \;\; 3 \;\; 6$
0 1 2 3
curr = 8

$j = 2$
$B[j] = 3$

$B[j+1] = 6$

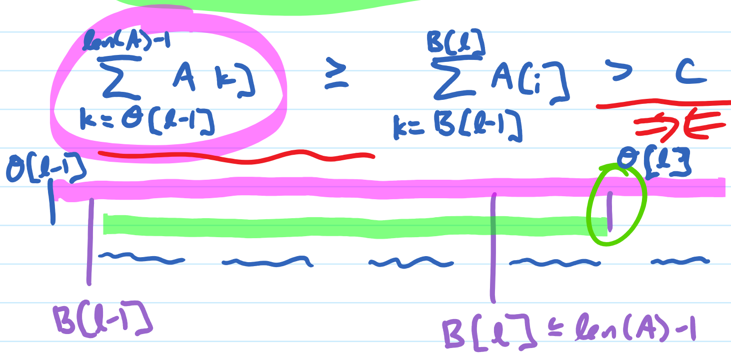
Maintenance (h): Suppose INV is true after i-1 iterations and $i < \text{len}(A)$

Suppose $\underbrace{\text{len}(\Theta)}_{\ell} \cancel{<} m_{new}$

$\text{len}(\Theta) \ge m_{old} = m_{new} - 1$
$\ell = m_{old}$ so $j = m_{old} - 1$ is in

then $B[\ell-1] \ge \Theta[\ell-1]$ INV h
$$\sum_{k=\Theta[\ell-1]}^{\text{len}(A)-1} A[k] \le c$$
Θ is valid
$$\sum_{k=B[\ell-1]}^{B[\ell]} A[i] > c$$
condition on C

$$\sum_{k=\Theta[\ell-1]}^{\text{len}(A)-1} A\,k] \ge \sum_{k=B[\ell-1]}^{B[\ell]} A[i] > c$$
$\Rightarrow \Leftarrow$ so $\text{len}(\Theta) \ge m_{new}$



We have $B[j] \geq \Theta[j]$ for $0 \leq j < m_{old}$ by INV h; need for $0 \leq j < m_{new}$ where $m_{new} = m_{old} + 1$. We don't change old values of B, so still true for $0 \leq j < m_{old}$; just need true for $j = m_{old}$ too

Suppose ~~$\Theta[m_{old}] > B[m_{old}]$~~

then $B[m_{old}-1] \geq \Theta[m_{old}-1]$ INV h

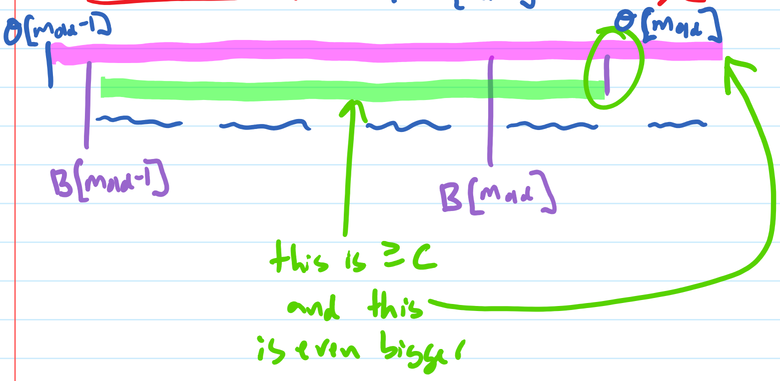
$$\sum_{k=\Theta[m_{old}-1]}^{\Theta[m_{old}]-1} A[k] \leq c$$ Θ is valid

$$\sum_{k=B[m_{old}-1]}^{B[m_{old}]} A[i] > c$$ condition on C

$$\sum_{k=\Theta[m_{old}-1]}^{\Theta[m_{old}]-1} A[k] \geq \sum_{k=B[m_{old}-1]}^{B[m_{old}]} A[i] > c$$

at least as many terms in 1st sum: $\Theta[m_{old}]-1 \geq B[m_{old}]$ and $\Theta[m_{old}-1] \leq B[m_{old}-1]$

$\Rightarrow\Leftarrow$ so $\Theta[m_{old}] \geq B[m_{old}]$

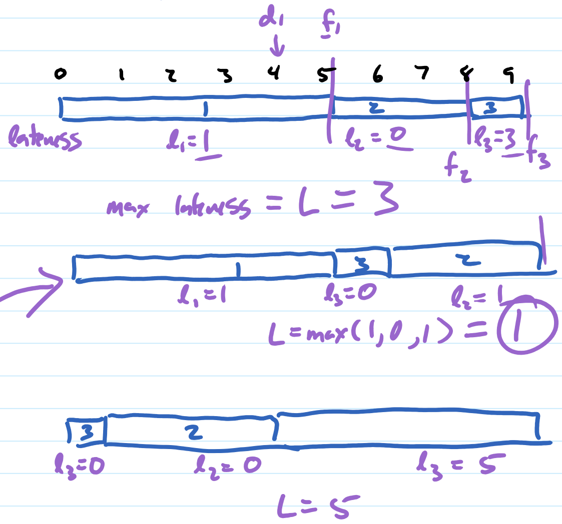


Minimizing Lateness

Given requests with lengths $t_1, \dots, t_n$, deadlines $d_1, \dots, d_n$,
find schedule that minimizes maximum lateness.

Ex:

i	t_i	d_i
1	5	4
2	3	8
3	1	6



max lateness = $L = 3$

$L = \max(1, 0, 1) = \boxed{1}$

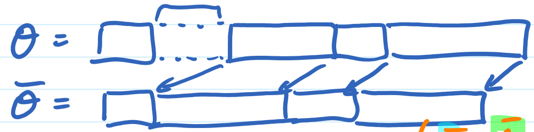
$L = 5$

shortest first

most ~~pressing~~ first ?

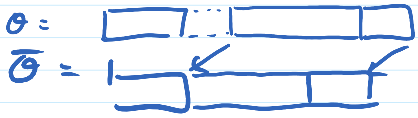
earliest deadline first

1) There is an optimal solution w/ no idle time



$\max(\bar{l}_1, \bar{l}_2, \dots, \bar{l}_n) \le \max(l_1, l_2, \dots, l_n)$

1) There is an optimal schedule with no idle time



$\bar{f}(i) = f(i) -$ idle time before i in schedule for all i

so $\bar{f}(i) \le f(i)$

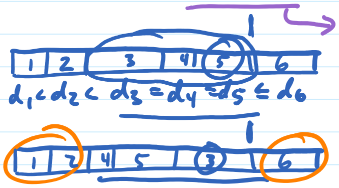
so $\bar{l}_i = \bar{f}(i) - d_i \le f(i) - d_i = l_i$

and $\bar{L} = \max_i(\bar{l}_i) \le \max_i(l_i) = L$

so $\bar{\mathcal{O}}$ is still optimal

2) All schedules with no idle time and no inversions have same max lateness

inversions → task i before task j but $d_i > d_j$



$d_1 < d_2 < d_3 = d_4 = d_5 \le d_6$

$\max(l_3, l_4, l_5) = l_5 = f(5) - d_5$

$\max(\bar{l}_3, \bar{l}_4, \bar{l}_5) = \bar{l}_3 = \bar{f}(3) - d_3$
$= f(5) - d_5$
$= l_5 = \max(l_3, l_4, l_5)$

$L = \max(l_1, l_2, l_3, l_4, l_5, l_6)$
$= \max(l_1, l_2, l_6, \max(l_3, l_4, l_5))$
$= \max(\bar{l}_1, \bar{l}_2, \bar{l}_6, \max(\bar{l}_3, \bar{l}_4, \bar{l}_5))$
$= \max(\bar{l}_1, \dots, \bar{l}_6) = \bar{L}$

$\max(\bar{l}_3, \bar{l}_4, \bar{l}_5) = l_3 = f(3) - d_3$

$= f(5) - d_5$

$= l_5 = \max(l_3, l_4, l_5)$

$= \max(\bar{l}_1, \bar{l}_2, \bar{l}_6, \max(\bar{l}_3, \bar{l}_4, \bar{l}_5))$

$= \max(\bar{l}_1, \ldots, \bar{l}_6) = \bar{L}$

3) There is a optimal schedule with no inversions, no idle time.

So greedy has no inversions, no idle time:

There is optimal Θ w/ no inversions, no idle (3)

greedy has same max lateness as Θ (2)

$\therefore$ greedy is optimal

same max lateness

Suppose not - that is, all optimal Θ with no idle time have inversions

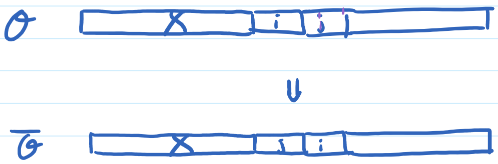
Pick optimal Θ with fewest inversions and no idle time (well-ordering) (1)

Let i, j be an inversion in Θ so i before j but $d_i > d_j$

Assume WLOG that j follows i immediately in Θ

(let x be index of i in Θ so $\Theta_x = i$
let y be index of j in Θ so $\Theta_y = j$
can't have $d_{\Theta_x} \le d_{\Theta_{x+1}} \le \ldots \le d_{\Theta_y}$
since then $d_i = d_{\Theta_x} \le d_{\Theta_y} = d_j$ and
i, j is not an inversion)

Construct $\bar{\Theta}$ by swapping (exchanging) i and j:



(still no idle time)

"monus"

$$x \mathbin{\dot{-}} y = \begin{cases} 0 & \text{if } y > x \\ x - y & \text{otherwise} \end{cases}$$

if $a > b$ then $x - a < x - b$ and $x \mathbin{\dot{-}} a \le x \mathbin{\dot{-}} b$

$l_j = f(j) \mathbin{\dot{-}} d_j$

$$l_i = f(i) \mathbin{\dot{-}} d_i \le f(j) \mathbin{\dot{-}} d_i \le \underline{f(j) \mathbin{\dot{-}} d_j} = l_j$$

def lateness | $f(j) > f(i)$ since j after i | $d_i > d_j$ since inverted | def lateness

so $\max(l_i, l_j) = l_j$

lateness of j went down since it was moved up

$$\bar{l}_j = \bar{f}(j) \mathbin{\dot{-}} d_j \le f(j) \mathbin{\dot{-}} d_j = l_j \quad \text{so } \bar{l}_j \le l_j$$

def lateness | $\bar{f}(j) \le f(j)$ since j moved up | def lateness

new lateness of i not worse than lateness of j

$$\bar{l}_i = \bar{f}(i) \mathbin{\dot{-}} d_i = f(j) \mathbin{\dot{-}} d_i \le f(j) \mathbin{\dot{-}} d_j = l_j$$

def lateness | $\bar{f}(i) = f(j)$ b/c | $d_i > d_j$ | def lateness

$$f(j) = \sum_{x \in X} t_x + t_i + t_j$$
$$\shortparallel$$
$$\bar{f}(i) = \sum_{x \in X} t_x + t_j + t_i$$

finish = sum of times of tasks before + own time

$$\max(\bar{l}_i, \bar{l}_j) \leq \max(l_j, l_j) = l_j \leq \max(l_i, l_j)$$

$$\max(\bar{l}_1, \ldots, \bar{l}_n) = \max\left(\max_{k \neq i,j} \bar{l}_k, \max(\bar{l}_i, \bar{l}_j)\right)$$

$$= \max\left(\max_{k \neq i,j} l_k, \max(\bar{l}_i, \bar{l}_j)\right)$$

lateness of tasks other than i,j didn't change

$$\leq \max\left(\max_{k \neq i,j} l_k, \max(l_i, l_j)\right)$$

max of larger things is larger

$$= \max(l_1, \ldots, l_n)$$

$\max(l_1, \ldots, l_n) \leq \max(\bar{l}_1, \ldots, \bar{l}_n)$ Θ is optimal

$\max(\bar{l}_1, \ldots, \bar{l}_n) = \max(l_1, \ldots, l_n)$ $\geq$ and $\leq$

$\therefore$ $\bar{\Theta}$ is optimal same max lateness as opt

$\bar{\Theta}$ has fewer inversions than Θ fixing i,j makes count ↓ 1

$\Rightarrow\Leftarrow$

so there is an optimal Θ with no inversions and no idle time