classmate

Date
Page

Discrete Fourier Transform

Def: The nth root of unity is a complex no. st. $x^n = 1$

Fact: The nth roots of unity are $\omega^0, \omega, \ldots \omega^{n-1}$

where

$$\omega^k = e^{2\pi i k/n}$$

$$(\omega^k)^n = \left(e^{2\pi i k/n}\right)^n = e^{2\pi i k}$$
$$= \cos(2\pi k) + i\sin(2\pi k)$$
$$= 1$$

Given polynomial

$a_0 + a_1 x + \ldots + a_{n-1} x^{n-1}$

we evaluate it at.

$\omega^0, \omega^1, \omega^2 \ldots \omega^{n-1}$

$$\begin{bmatrix} y_0 \\ | \\ | \\ | \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \ldots & \\ 1 & \omega & \omega^2 & \ldots & \\ 1 & \omega^2 & \omega^4 & \ldots & \\ | & & & & \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \omega^{3(n-1)} & \ldots \end{bmatrix} \begin{bmatrix} a_0 \\ | \\ | \\ | \\ a_{n-1} \end{bmatrix}$$

→ Fast Fourier Transform

Goal → Evaluate degree $(n-1)$ pol at roots of unity $\omega^0 \ldots \omega^{n-1}$

Divide → Break polynomial into even & odd powers.

$A_{even}(x) = a_0 + a_2 x + a_4 x^2 + \ldots a_{n-2} x^{n/2 - 1}$

$A_{odd}(x) = a_1 + a_3 x + \ldots \ldots + a_{n-1} x^{n/2 - 1}$

Conquer Evaluate

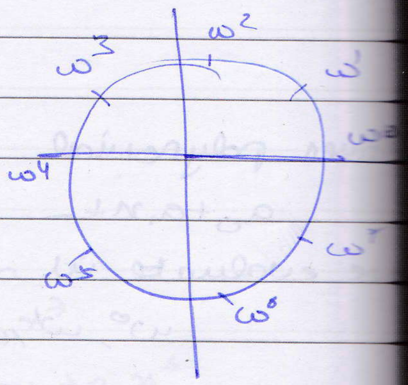
Evaluate A_{even} & A_{odd} at $n/2$ th roots of unity

$$r^0, r^1, r^2 \ldots r^{n/2-1}$$

Combine

Evaluate at ω^k & $\omega^{k+n/2}$ $n=8$

$k = 0 \ldots n/2 - 1$



$\omega^0 \ldots \omega^{n-1}$

$$A(\omega^k) = A_{even}((\omega^k)^2) + \omega^k A_{odd}((\omega^k)^2)$$

$$A(\omega^{k+n/2}) = A_{even}((\omega^{k+n/2})^2) + \omega^{k+n/2} A_{odd}((\omega^{k+n/2})^2)$$

$$A(x) = A_{even}(x^2) + x\, A_{odd}(x^2)$$
$$A(-x) = A_{even}(x^2) - x(A_{odd}(x^2))$$

$$(\omega^{k+n/2})^2$$
$$= \omega^{2k+n}$$
$$= (e^{2\pi i/n})^{2k+n}$$

$$= (e^{2\pi i/n})^{2k} \cdot e^{2\pi i}$$
$$= \omega^{2k}$$

classmate

Date

Page

$$\omega^{k+n/2} = \omega^{k} \cdot \omega^{n/2}$$

$$= \omega^{k} \cdot e^{2\pi i/n \cdot n/2}$$

$$= \omega^{k} \cdot e^{\pi i}$$

$$= -\omega^{k} \quad (= -1)$$

$$A(\omega^{k}) = A_{even}(\omega^{2k}) + \omega^{k}\{A_{odd}(\omega^{2k})\}$$

$$A(\omega^{k+n/2}) = A_{even}(\omega^{2k}) - \omega^{k} A_{odd}(\omega^{2k})$$

eg. $A(x) = a_0 + a_1 x + \dots + a_{n-1} x^{n-1}$

pt's $\omega^0, \omega^1 \dots \omega^{n-1}$

take $n=4$

$\omega^0, \omega^1, \omega^2, \omega^3$

FFT divide & conquer $k = 0, \dots n/2 - 1$ $(\nu^0 \dots \nu^{n/2-1})$

$k=0$

$$A(\omega^0) = A_{even}(\omega^0) + \omega^0 A_{odd}(\omega^0)$$

$$A(\omega^{0+2}) = A_{even}(\omega^0) - \omega^0 A_{odd}(\omega^0)$$

$k=1$

$$A(\omega^1) = A_{even}(\omega^2) + \omega A_{odd}(\omega^2)$$

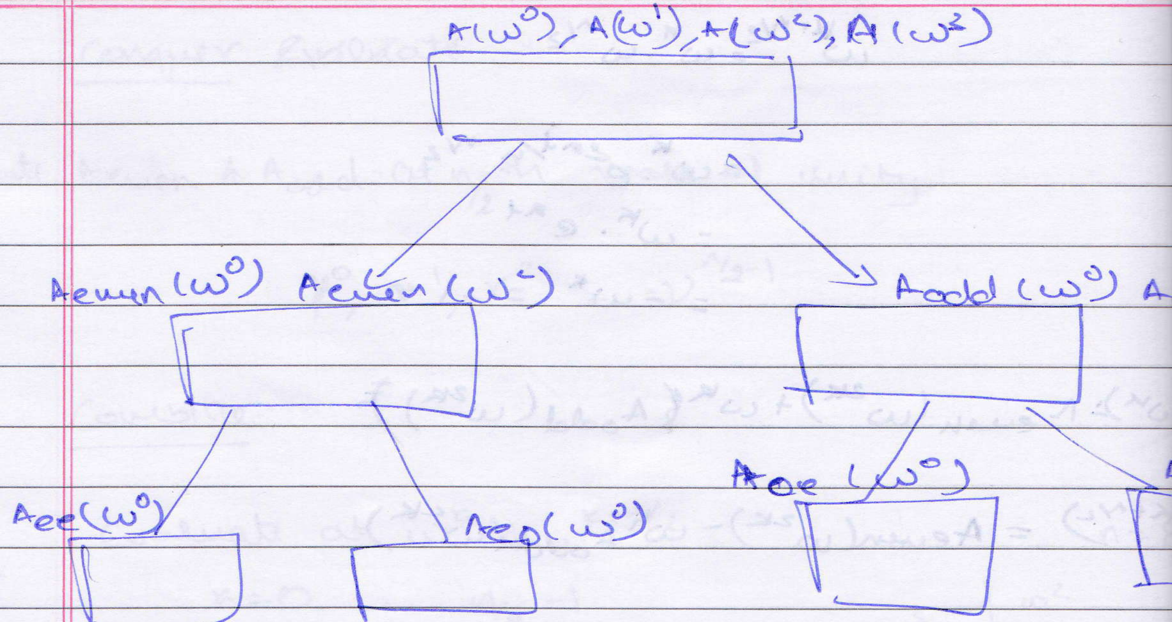
$$A(\omega^{1+2}) = A_{even}(\omega^1) - \omega A_{odd}(\omega^2)$$

$A_{even}(\omega^0)$, $A_{even}(\omega^2)$

$A_{odd}(\omega^0)$, $A_{odd}(\omega^2)$

$A_{ee}(\omega^0)$ $A_{eo}(\omega^0)$

$A_{oe}(\omega^0)$ $A_{oo}(\omega^0)$



FFT (coeff ⇒ Pt. value)

FFT$(n, a_0, a_1 \ldots a_{n-1})$

$(e_0, e_1 \ldots e_{n/2-1}) = \text{FFT}(n/2, a_0, a_2 \ldots a_{n-2})$ even

$(f_0, f_1 \ldots f_{n/2-1}) = \text{FFT}(n/2, a_1, a_2 \ldots a_{n-1})$ odd

for $k=0$ to $n/2-1$

$\omega^k = e^{2\pi i k/n}$

$y_k = e_k + \omega^k f_k$

$y_{k+n/2} = e_k - \omega^k f_k$

Return $(y_0, \ldots y_{n-1})$

Point value $\Rightarrow$ coeff

$$\begin{bmatrix} y_0 \\ | \\ | \\ | \\ | \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots \\ 1 & x_1 & x_1^2 & \cdots \\ & & & \\ & & & \\ 1 & x_{n-1} & x_{n-1}^2 & \cdots \end{bmatrix} \begin{bmatrix} a_0 \\ | \\ | \\ | \\ a_{n-1} \end{bmatrix}$$

a_0 till a_{n-1}

$$\begin{bmatrix} a_0 \\ | \\ | \\ a_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots \\ & & & \\ & & & \end{bmatrix}^{-1} \begin{bmatrix} y_0 \\ | \\ | \\ y_{n-1} \end{bmatrix}$$

Matrix Inverse
(LU Fact
$O(n^3)$)

fast mult
$O(2^{.37})$)

Inverse FFT

$x_k = \omega^k$, $\omega^0, \omega^1 \cdots \omega^{n-1}$

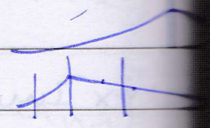
$$\begin{bmatrix} a_0 \\ | \\ | \\ | \\ a_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots \\ 1 & \omega^1 & \omega^2 & \omega^3 & \cdots \\ 1 & \omega^2 & \omega^4 & \omega^8 & \cdots \\ 1 & \omega^3 & \omega^9 & & \cdots \\ 1 & \omega^{(n-1)} & \omega^{2(n-1)} & \omega^{3(n-1)} & \cdots \end{bmatrix}$$

$$\begin{bmatrix} a_0 \\ \vdots \\ \vdots \\ a_{n-1} \end{bmatrix} = \frac{1}{n} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/\omega & 1/\omega^2 & 1/\omega^3 \\ 1 & 1/\omega^2 & 1/\omega^4 & 1/\omega^6 \\ \vdots & & & \\ 1 & \frac{1}{\omega^{n-1}} & \frac{1}{\omega^{2(n-1)}} & \frac{1}{\omega^{3(n-1)}} \end{bmatrix} \begin{bmatrix} y_0 \\ \vdots \\ y_{n-1} \end{bmatrix}$$

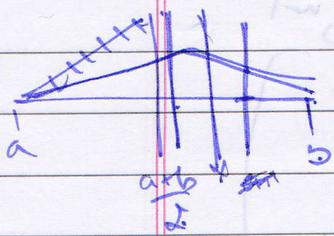
$\boxed{\omega^{-k} = e^{-2\pi i k/n}}$ Inverse FFT

Q: Unimodal search,

An array A[1, - - n] is unimodal if it consists of increasing sequence followed by a dec seq. Give an algorithm to compute Max element in A O(logn).



```
a=1 , b=n
while a<b
    mid= A[(a+b)/2]
    if A[mid] < A[mid+1]
        a = mid+1
    else
        b = mid
return A[a]
```



$T(n) = T(n/2) + O(1)$

$= O(\log n)$

Algo:-

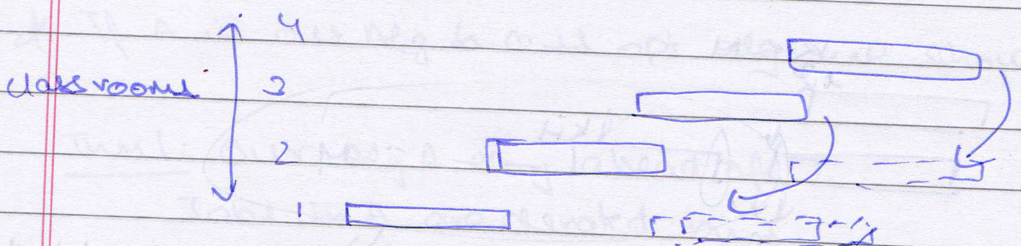
Interval_sch$(n, s_1 - s_n, f_1 - f_n)$
sort jobs by finish time || $O(n \log n)$
$f_1 \leq \ldots \leq f_n$
$A \leftarrow \phi$
for $j = 1$ to n
$O(n)$ || if j is compatible with A] → s_j compare finish time of last job
$A \leftarrow A \cup \{j\}$
Return A

$T(n) = O(n \log n)$

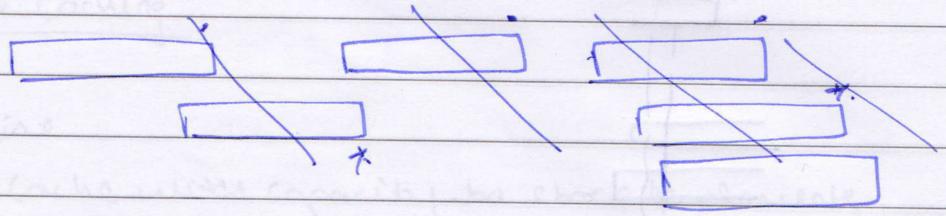
Interval Partitioning

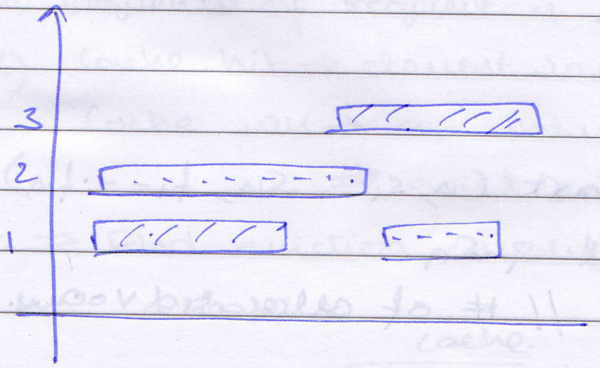
-- Intro

Lecture j starts at s_j & finishes at f_j.
Goal → schedule set of lectures to min # of class



1.) earliest start time
2.) earliest finish time.
3.) shortest interval
4.) least overlap/ fewer conflicts





✓EST — Algo —
EFT X
S.I X
OFC X

|| # of allocated room

-- Is earliest start time optimal.

&&i

let A be our Algo $\rightarrow i$

let O be opt Algo $\rightarrow j$

$s(i_r) < s(j_r)$ $\forall r \leq \max\{$jobs in A or O$\}$

Pf. base case $\rightarrow$ obvious

let this be true for $r = 1 \ldots k$.

$s(i_k) < s(j_k)$

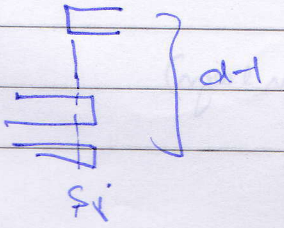
for k+1:

$s(i_{k+1}) < s(j_{k+1})$

Pf. ~~If min~~ Idea $\rightarrow$ If d is the min no. of classrooms then at any time d lectures should be held for the algo to be optimal.

* The lecture j is being assigned to dth class room.

$\Rightarrow (d-1)$ lectures are incompatible at s_j.



Pf. that Cashier's Alg is optimal for US coins 1, 5, 10, 25, 10[illegible]

$c_1, c_2 \cdots c_{k-1}$

Let we need to pay x s.t. $c_k \le x \le c_{k-1}$

K	c_k	All optimal solns should satisfy.	max value of $c_1 \ldots c_{k-1}$
1	1	1 cents ≤ 4	
2	5	5 cents ≤ 2	4
3	10	5 cents + 10 cents ≤ 2	9
4	25	25 cents ≤ 3	24
5	100	No limit	99.

Proof of interval scheduling.

Goal → given set of jobs, find the ones (max) which [illegible] compatible.

Alg → earliest finish time

* If A is our Alg & O is opt alg, we show $|A| = |O|$

Thm 1: Our Alg A stay ahead of O.

Jobs in A are denoted by i

" " O " " " j

$\forall r \le \max\{\text{jobs in A or O}\}$

$f(i_r) \le f(j_r)$

Date ______
Page ______

Induction Pf

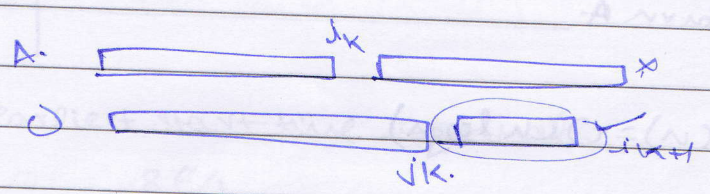
Base case → obvious.

Hypothesis → let this is true for. $r = 1, \ldots, k$.

$f(i_k) \le f(j_k)$

Induction step → we show.

$f(i_{k+1}) \le f(j_{k+1})$

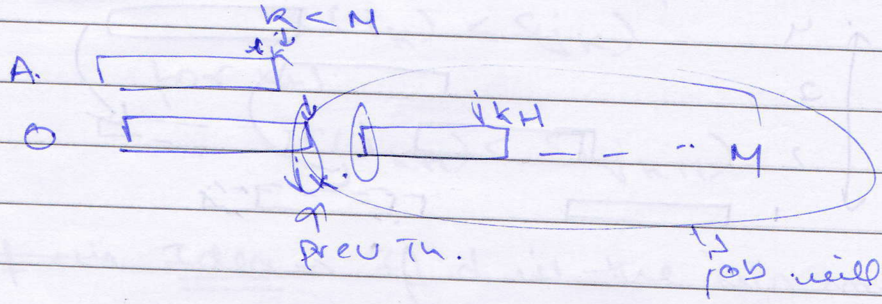


Thm 2 → The greedy Alg (shortest finish time) gives opt. no. of compatible jobs.

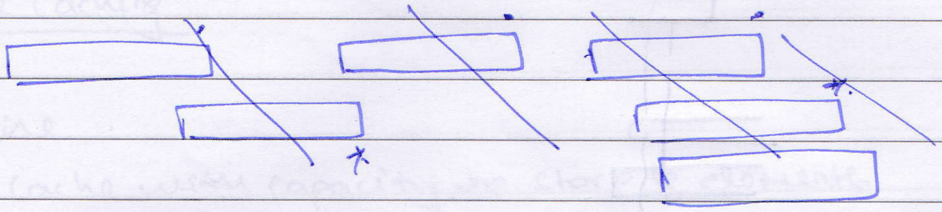
Pf. contradiction.

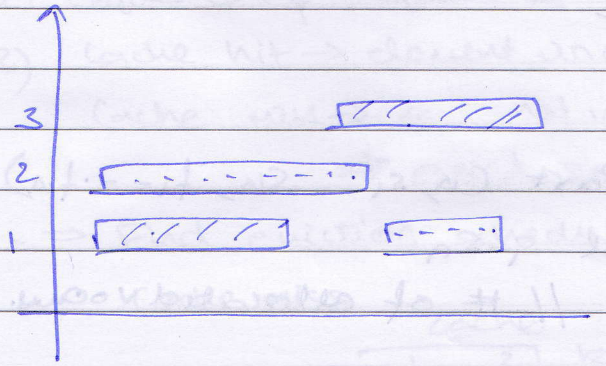
A → finds k jobs mut. compatible,

O → " O " " "



$|A| \geq k+1$





EST ✓
EFT X
S.I X
OFC X

-- Is earliest start time optimal.

&&

let A be our Algo → i

let O be opt Algo → j

$s(i_r) \le s(j_r)$ $\forall r \le \max\{\text{jobs in } A \text{ or } O\}$

Pf. base case → obvious

let this be true for $r = 1 \ldots k$.

$s(i_k) \le s(j_k)$

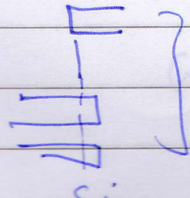
for k+1

$s(i_{k+1}) \le s(j_{k+1})$

Pf. ~~If min~~ Idea → If d is the min no. of classrooms then at any time d lectures should be held for the algo to be optimal.

* The lecture j is being assigned to dth class room.

⇒ (d-1) lectures are incompatible at s_j.



d-1

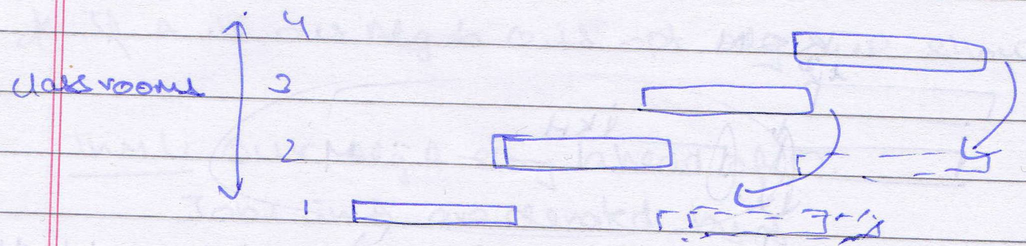
Algo:-

Interval_sch$(n, s_1 - s_n, f_1 - f_n)$
sort jobs by finish time || $O(n \log n)$
$f_1 \le \dots \le f_n$
$A \leftarrow \phi$
for $j = 1$ to n
$O(n)$ || if j is compatible with A] → s_j compare w finish time last job in
$A \leftarrow A \cup \{j\}$
Return A

$T(n) = O(n \log n)$

Interval Partitioning

-- Intro
Lecture j starts at s_j & finishes at f_j.
Goal → schedule set of lectures to min # of class



1) earliest start time
2) earliest finish time.
3) shortest interval
4) least overlap / fewer conflicts

b). 80, 60, 60

$\downarrow$ greedy $\quad \downarrow$ optimal $\quad n=120$

$$G = \left\{ \frac{n+1}{2}, \frac{n}{2}, \frac{n}{2} \right\}$$

$$\lim_{n \to \infty} \frac{n+1}{2n} \to \frac{1}{2}$$

a). PROVE → SEATadmits all groups of size ≤ n or it admits atleast $n/2$ people.

Case I size of max group (S) $\geq n/2$.

Case II. size of max group (S) $< n/2$.

$a, b, c, d \quad \leq n/2$

$a+b+c < n/2$

$a+b+c+d > n/2$

keep adding till it exceeds n.

$a+b+c+d - \cdots + l > n$

$l < n$

$> n/2$

$< n/2$

→ n.. ~~to n~~

K. → max. group size $< n/2$.

Part-2 Design of Algos

*A. GREEDY TYPE

→ Coin changing
→ Int sch
→ Int Partitioning
→ Minimize Lateness
→ Optimal caching

(scheduling)

→ Single Link Clustering
→ Dijkstra → $O(mn)$ → $O(m\log n)$ (Binary heap) | $O(n^2)$, $O(m+n\log n)$ (Fibonacci)

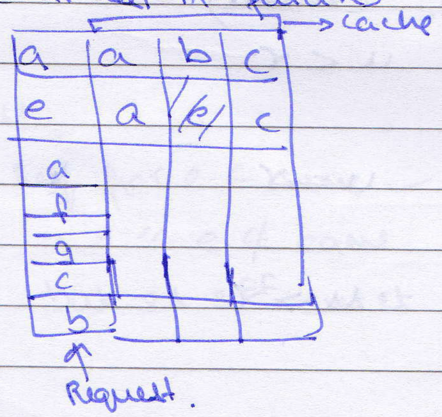
→ Min Spanning Tree
-- Prim
-- Kruskal
--

→ Min Cost Aborescences

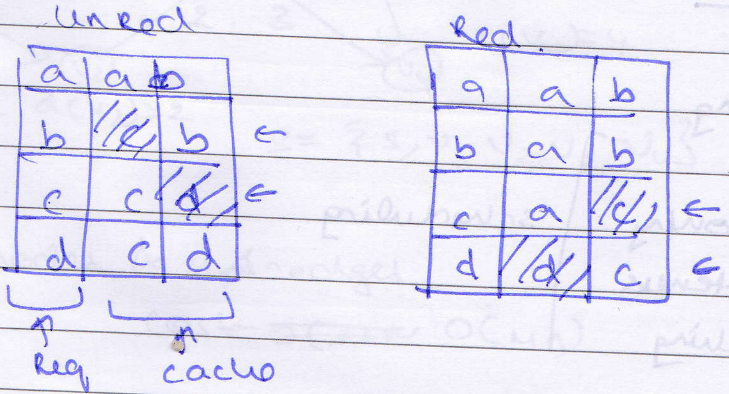
Graph

— Optimal Caching

-- Recall (Farthest-in-future)

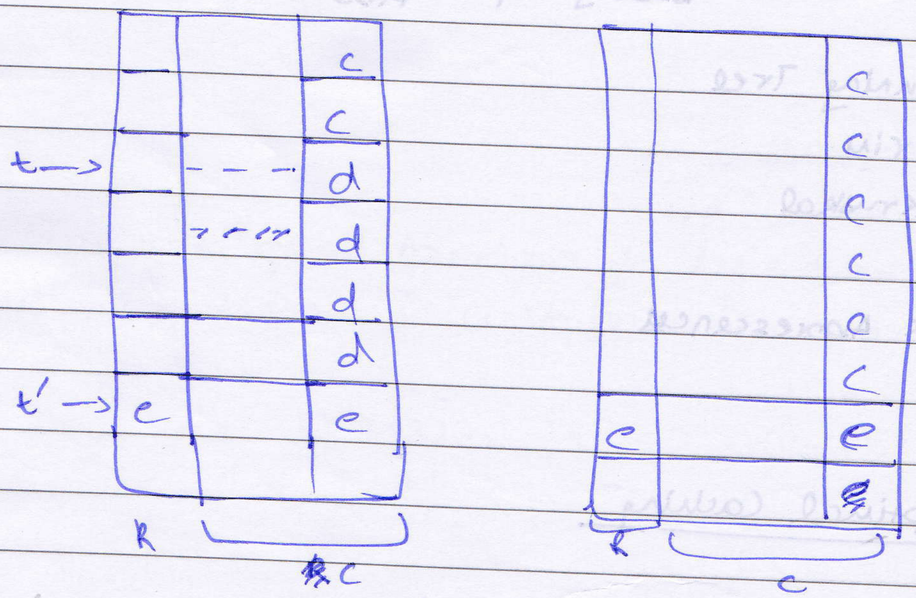


Thm 1 -> An unreduced schedule can be converted to reduced schedule with no more cache misses



(eviction before request)

Pf 1



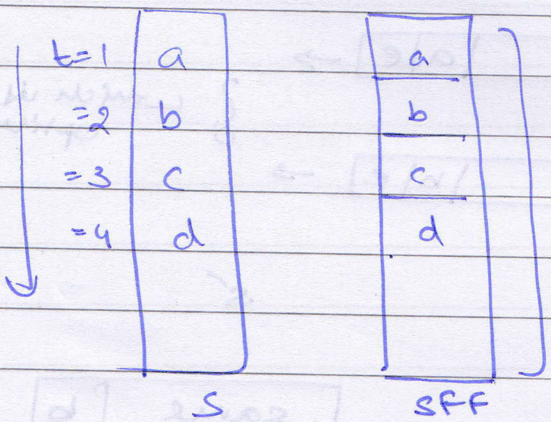
Case 1: $e = d$ $u < r$ (misses)

Case 2: $e \neq d$ $u = r$

Case 3: $e = c$ $u > r$

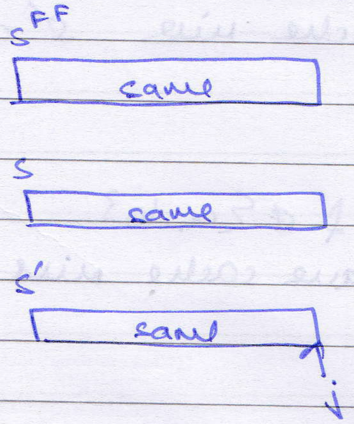
Thm2 → FF is optimal (min # of cache misses)

Invariant → There exists an optimal reduced schedule S that has same eviction schedule as S^{FF} for first j requests.



j=100 requests

Pf. I.P → If ∃ an optimal reduced schedule S that has same eviction as S^{FF}, then there exist an optimal reduced schedule S', that has same eviction such as S^{FF} for (j+1) request.



(j+1)-th request

Case 1: Req for e ∈ same ✓

Case 2: " " e ∉ same

but S & S^{FF} evict same element ✓

PTO.