Fall-22 (Paging Math)

(2) a) RAM size = 2 GB = 2×2^{30} B.

Process " = 128 MB = 128×2^{20} B = $2^7 \times 2^{20}$ B

Frame size = page size = 128 KB = $2^7 \times 2^{10}$ B ; offset = 17 Bits.

$$\text{No. of Frames} = \frac{\text{RAM size}}{\text{Frame size}} = \frac{2^{31}\ B}{2^{17}\ B} = 2^{14}\ B = 14 \text{ Bits}$$

$$\text{No. of pages} = \frac{\text{Process size}}{\text{Page size}} = \frac{2^{27}\ B}{2^{17}\ B} = 2^{10}\ B = 10 \text{ Bits}$$

Given frame location = 105th. and process location = 330.

Physical address format = | 14 Bits | 17 Bits |

logical address " = | 10 Bits | 17 Bits |

Here, physical address = | 00000001101001 | 00000000000011101 | [$\because (105)_{10} = (1101001)_2$]

logical address = | 0101001010 | 00000000000011101 | [$\because (330)_{10} = (101001010)_2$]

Spring-2022

(3) a) Given, Ram size = 4 GB = $2^2 \times 2^{30} = 2^{32}$ B

Process " = 256 MB = $2^8 \times 2^{20} = 2^{28}$ B.

(s) Avg process size = 32 MB = $2^5 \times 2^{20} = 2^{25}$ B.

(e) An entry page = 4 bytes = 2^2 Bytes.

We know, Frame size = Page size = $\sqrt{25e} = \sqrt{2 \times 2^{25} \times 2^2}$ B

$= \sqrt{2^{28}}$ B. $= 2^{14}$ B.

[Thus, offset = 14 bits].

~~Given,~~

No. of frame = $\frac{\text{Ram size}}{\text{frame size}} = \frac{2^{32}\ B}{2^{14}\ B} = 2^{18}\ B = 18$ bits

No. of page = $\frac{\text{Process size}}{\text{page size}} = \frac{2^{28}\ B}{2^{14}\ B} = 2^{14}\ B = 14$ bits

Given, Frame location = $(111111)_2$ [ques-24 10 তুল maybe]

page " = $(1010)_2$.

offset = $(5246)_{10} = (101000111111.10)_2$

Physical address format = | 18 bits | 14 bits |

Logical " " = | 14 bits | 14 bits |

Here, physical address = | 00000000000111111 | 0101000111111 0 |

logical address = | 00000000001010 | 01010001111110 |

Fall-2021

(4) a) RAM size = 32MB = $2^5 \times 2^{20}\ B = 2^{25}\ B$ ~~= 25 Bits~~.

Process " = 64MB = $2^6 \times 2^{20}\ B = 2^{26}\ B$ ~~= 26 Bits~~.

page size = frame size = 32KB = $2^5 \times 2^{10}\ B = 2^{15}\ B$. [offset = 15 Bits]

No of frames = $\frac{\text{RAM size}}{\text{frame size}} = \frac{2^{25}}{2^{15}}\ B = 2^{10}\ B$ = 10 Bits

No. of page = $\frac{\text{process size}}{\text{page size}} = \frac{2^{26}}{2^{15}}\ B = 2^{11}\ B = 11$ bits

Given, Frame location $(28)_{10} = (11100)_2$

page " $(130)_{10} = (10000010)_2$

offset = $(11110)_2$

physical address format = | 10 bits | 15 bits |

Logical " " = | 11 bits | 15 bits |

Here, physical address = | 0000011100 | 000000000011110 |

logical " = | 00010000010 | 000000000011110 |

Spring-2023

(2) a) RAM size = 8GB = $2^3 \times 2^{30} = 2^{32}$ B

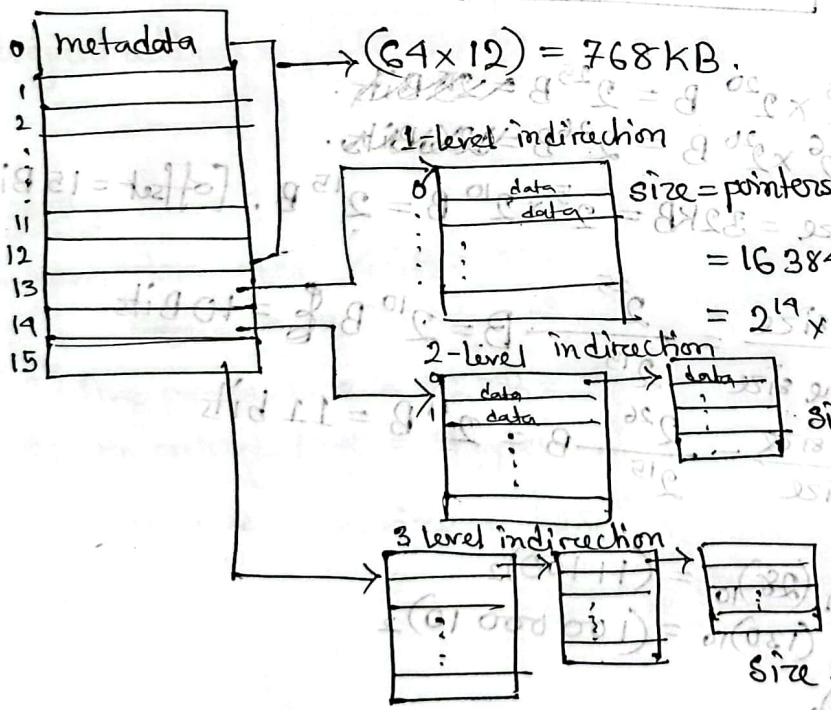
Process size = 1GB = $1 \times 2^{30} = 2^{30}$ B.

Avg process size =

Spring-2023

(2) c). Given, ~~64KB~~ Block size = 64KB = $2^6 \times 2^{10} = 2^{16}$ B.

pointer size = 4B = 2^2 B.



$(64 \times 12) = 768$KB.

pointers = $\frac{2^{16}\text{KB}}{2^2\text{B}}$

$= 2^{14}$

$= 16384$.

size = pointers × block size

$= 16384 \times 2^{16}$ B

$= 2^{14} \times 2^{16}$ B $= 2^{30}$ B $= 1$GB.

size $= 16384 \times 16384 \times 2^{16}$ B

$= 2^{14} \times 2^{14} \times 2^{16}$ B

$= 2^{44} = 2^4$ TB $= 16$TB

size $= 16384 \times 16384 \times 16384 \times 2^{16}$

$= 2^{14} \times 2^{14} \times 2^{14} \times 2^{16}$ B

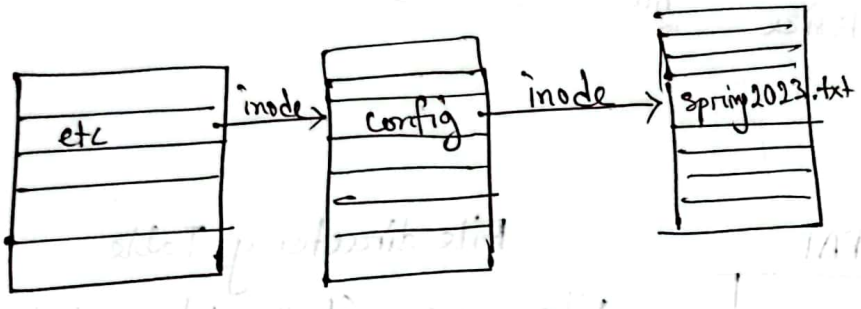
$= 2^{58}$ B $= 2^8 \times 2^{50} = 2^8$ PB

$= 256$PB

~~size limit = 7~~

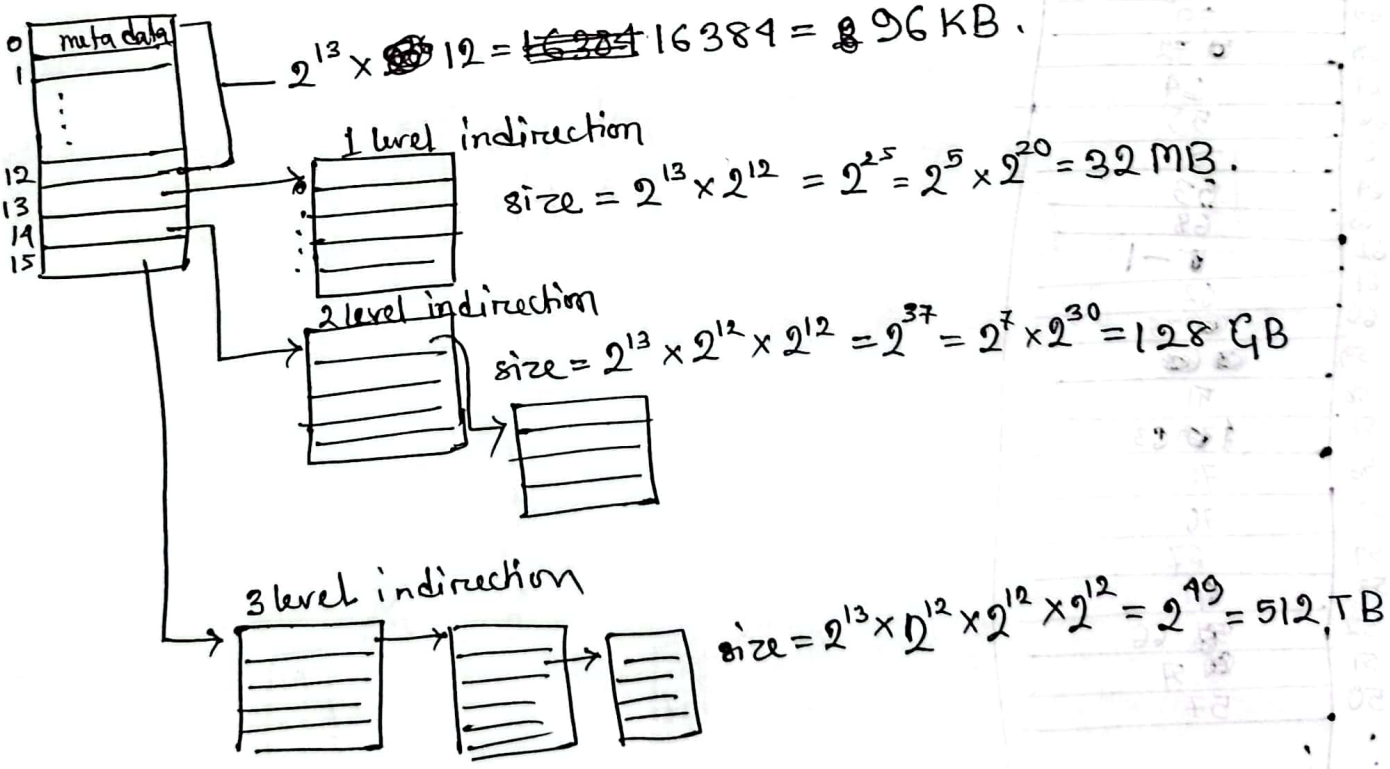
$$\text{Size} = 768\,KB + 1\,GB + 16\,TB + 256\,PB$$
$$= 768\,KB + 2^{20}\,KB + (16 \times 2^{30})\,KB + (256 \times 2^{40})\,KB$$
$$= 2.81 \times 10^{14}\,KB.$$

File location mechanism:



Fall-2022

(2) c) Block size = $8\,KB = 2^3 \times 2^{10}\,B = 2^{13}\,B$
pointer size = 2 B.
pointer no. = $\frac{2^{13}}{2} = 2^{12}$.



$2^{13} \times 12 = 16384 = 96\,KB.$

1 level indirection
size = $2^{13} \times 2^{12} = 2^{25} = 2^5 \times 2^{20} = 32\,MB.$

2 level indirection
size = $2^{13} \times 2^{12} \times 2^{12} = 2^{37} = 2^7 \times 2^{30} = 128\,GB$

size = $2^{13} \times 2^{12} \times 2^{12} \times 2^{12} = 2^{49} = 512\,TB$

size limit = $96\,KB + 32\,MB + 128\,GB + 512\,TB.$
$= 96\,KB + (32 \times 2^{10})\,KB + (128 \times 2^{20})\,KB + (512 \times 2^{30})\,KB$
$= \sim\sim$

FAT

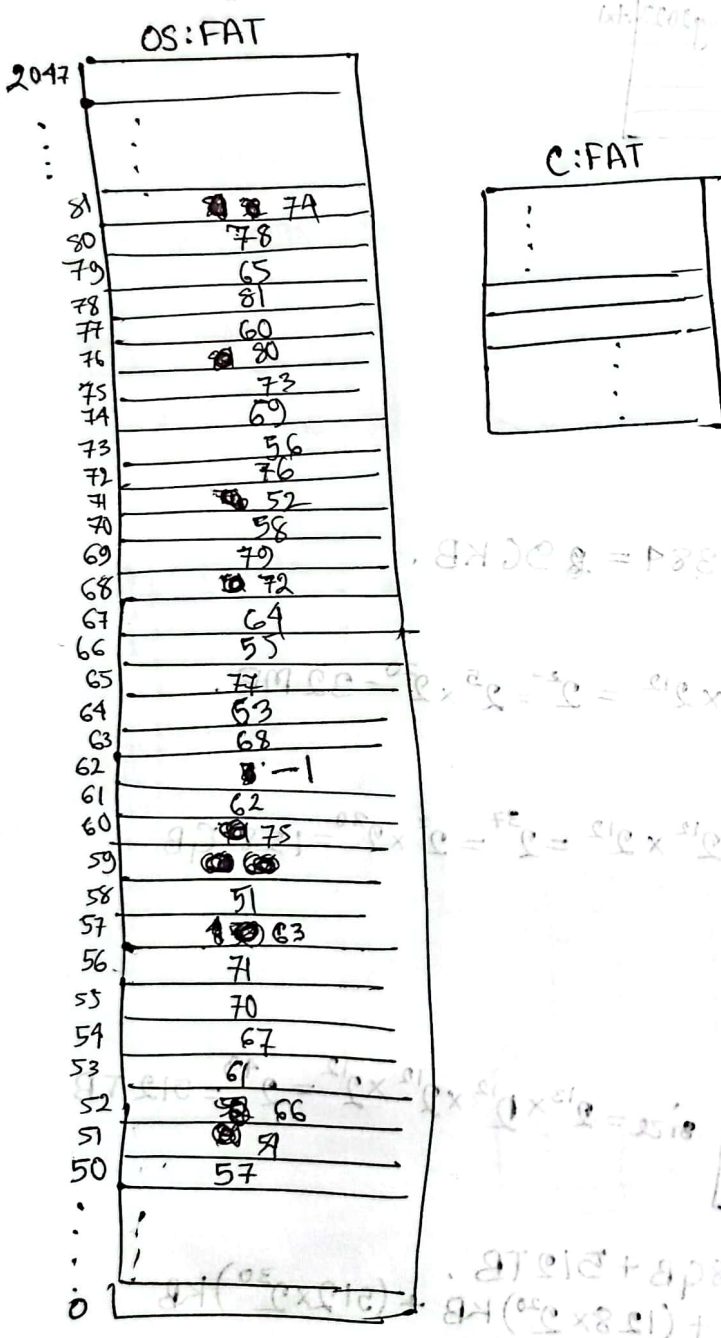
~~Spring~~ Fall-2022

② g) Block (in disk) = 2048.

Each block = 64 KB = $2^6 \times 2^{10}$ B

File size = 2 MB = 2×2^{20} B.

$$\text{No. of blocks} = \frac{\text{File size}}{\text{each block size}} = \frac{2^{21}}{2^{16}} = 2^5 = 32$$



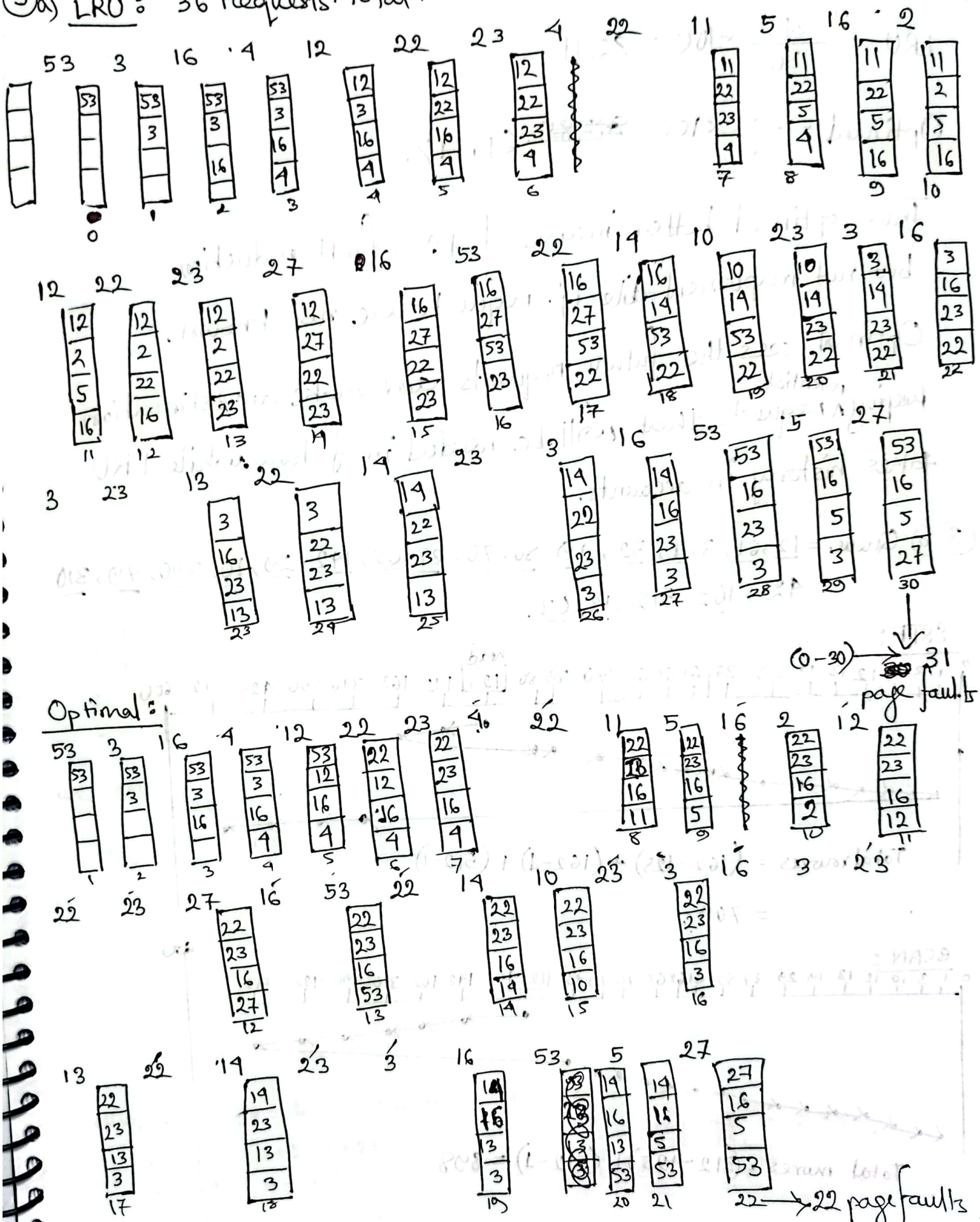
C:FAT

File directory: Table

File name	Starting point	metadata
C:	1	
OS	10	
data.text	50	

Fall-2022

(3)a) LRU: 36 requests. total.



Comparision:

LRU = $\frac{31}{36} \times 100 = 86.11$

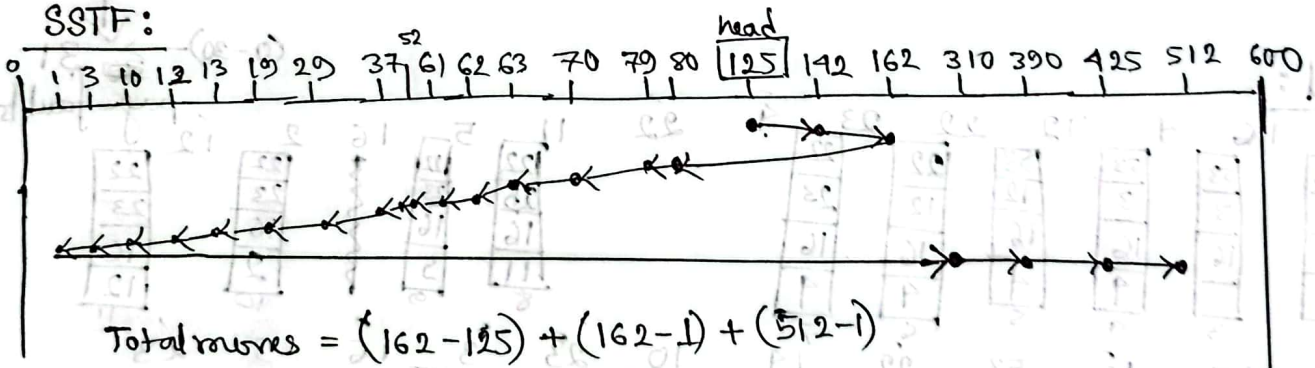
Optimal = $\frac{22}{36} \times 100 = 61.11\%$.

Thus optimal better incase of page fault reduction but not implementable if requests are not known.

Optimal sees the future requests and make an estimation keeping potential requests that will be needed in future while LRU takes history in accounts.

(3) b) Queue = 12, 61, 3, 1, 52, 19, 80, 70, 37, 62, 142, 29, 10, 390, 79, 310, 425, 162, 512, 13, 63.

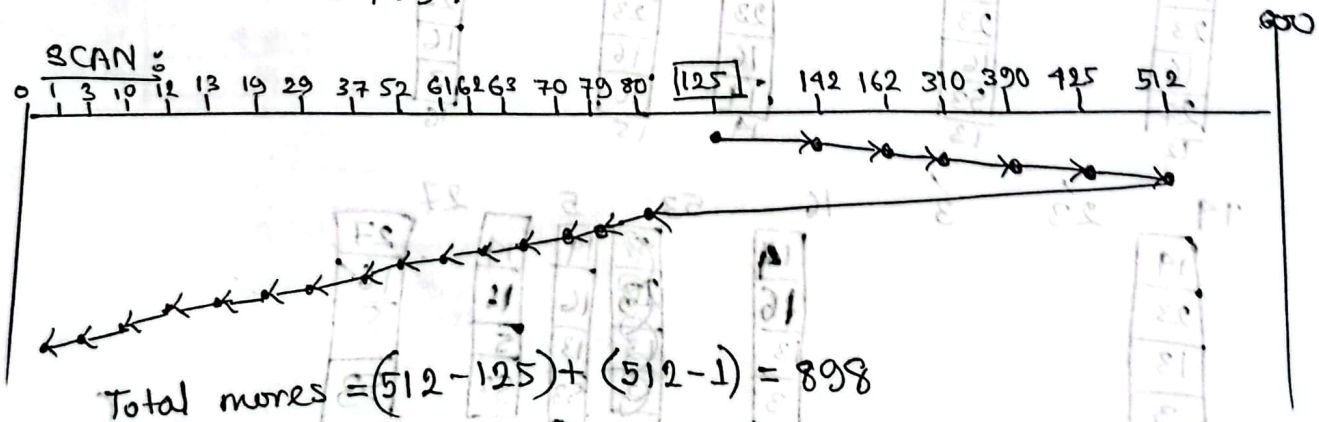
SSTF:



Total moves = (162 − 125) + (162 − 1) + (512 − 1)

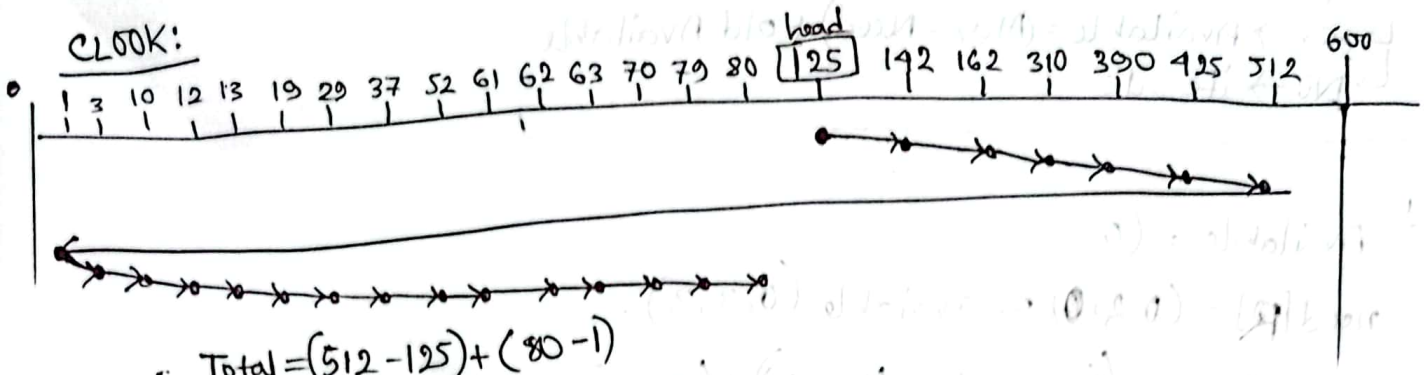
= 709.

SCAN:



Total moves = (512 − 125) + (512 − 1) = 898

CLOOK:



- Total = (512 − 125) + (80 − 1)

  = 466.

CLOOK is better.

Bankers Algo:

Spring 22.

(4) a) Given, Resources = {A: 10, B: 14, Z: 7}

	Allocation		
	A	B	Z
P	3	3	0
Q	2	0	0
R	3	0	2
D	2	1	1
S	0	1	1

	MAX		
	A	B	Z
P	7	3	2
Q	3	2	2
R	3	3	2
D	4	2	3
S	3	3	3

we know, Need = ~~Allo~~ Max − Allocation.

So,

	Need		
	A	B	Z
P	4	0	2
Q	1	2	2
R	0	3	0
D	2	1	2
S	3	2	2

also, Available = Total resource − Allocated

A = 10 − (3+2+3+2+0) = 0

B = 14 − (3+1+1) = 3

C = 7 − (2+1+1) = 3

Thus

Available		
A	B	C
0	3	3

** Check if Available থেকে Need কম

↳ yes → Available = (Max - Need) + old Available

↳ NO → iterate