

Assignment 2 :

Process	Allocation			max	Available
P0	A1	B3	C5	1 6 5	6 5 6
P1	1	0	0	2 1 3	
P2	2	0	0	3 4 1	
P3	4	1	1	4 5 7	
P4	5	4	3	5 0 0	

i) Total Amount : A  $\Rightarrow$  19  
                          B  $\Rightarrow$  13  
                          C  $\Rightarrow$  16

	A	B	C
P0	0	3	0
P1	1	1	3
P2	1	4	5
P3	0	4	6
P4	0	-4	-3

iii) Yes, the system is safe.  
      $\Rightarrow$  Safe Sequence: P0 P1 P2 P3 P4

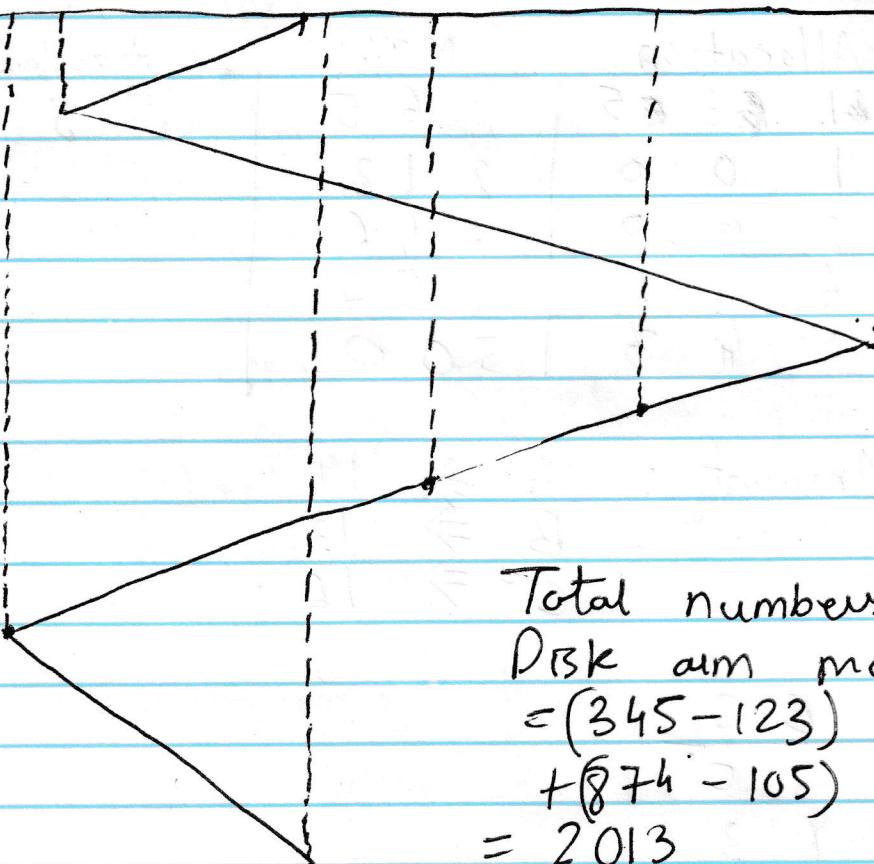
Process	Allocation			max	need	available
	A	B	C			
P0	1	3	5	1 6 5	0 3 0	2 0 3
P1	5	5	3	2 1 3	-3 -4 0	
P2	2	0	1	3 4 6	1 4 5	
P3	4	1	1	4 5 7	0 4 6	
P4	5	4	3	5 0 0	0 -4 -3	

State : Safe  
Safe Sequence : P1 P2 P3 P4 P0

Q2)

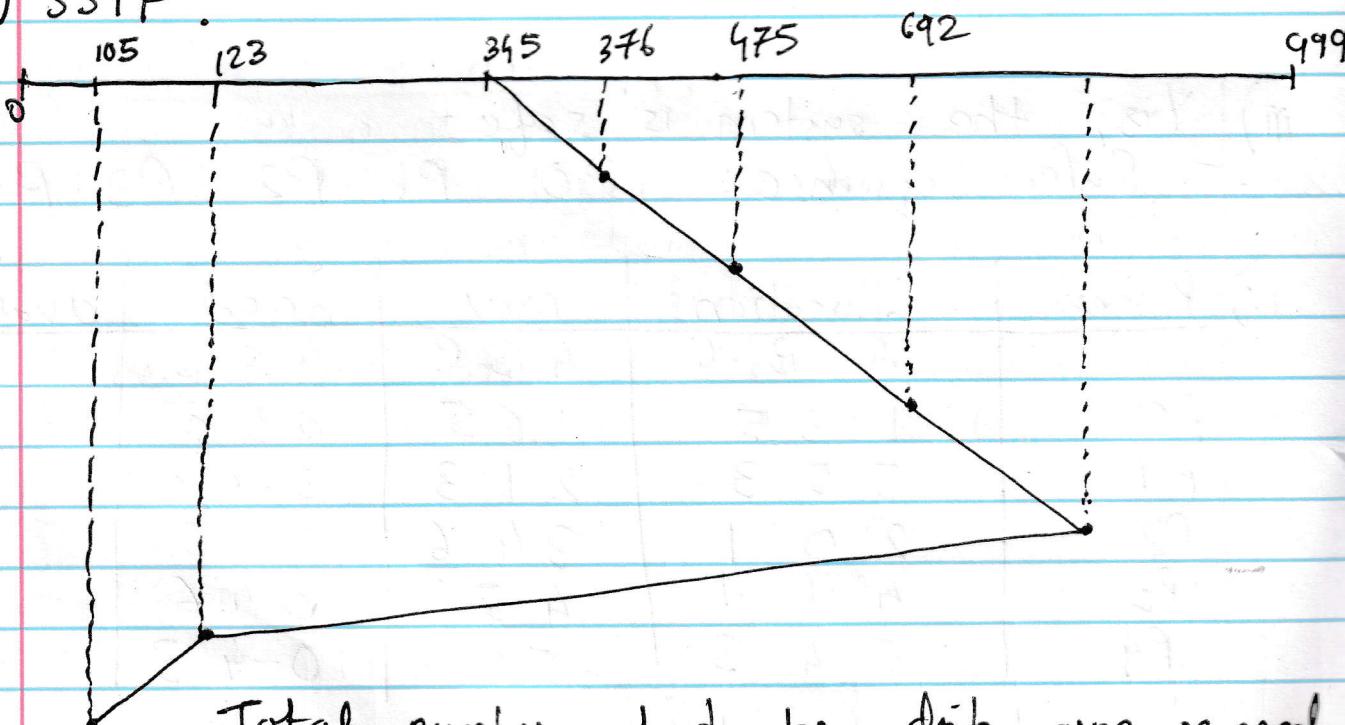
i)

105 123 345 376 475 692 874 999



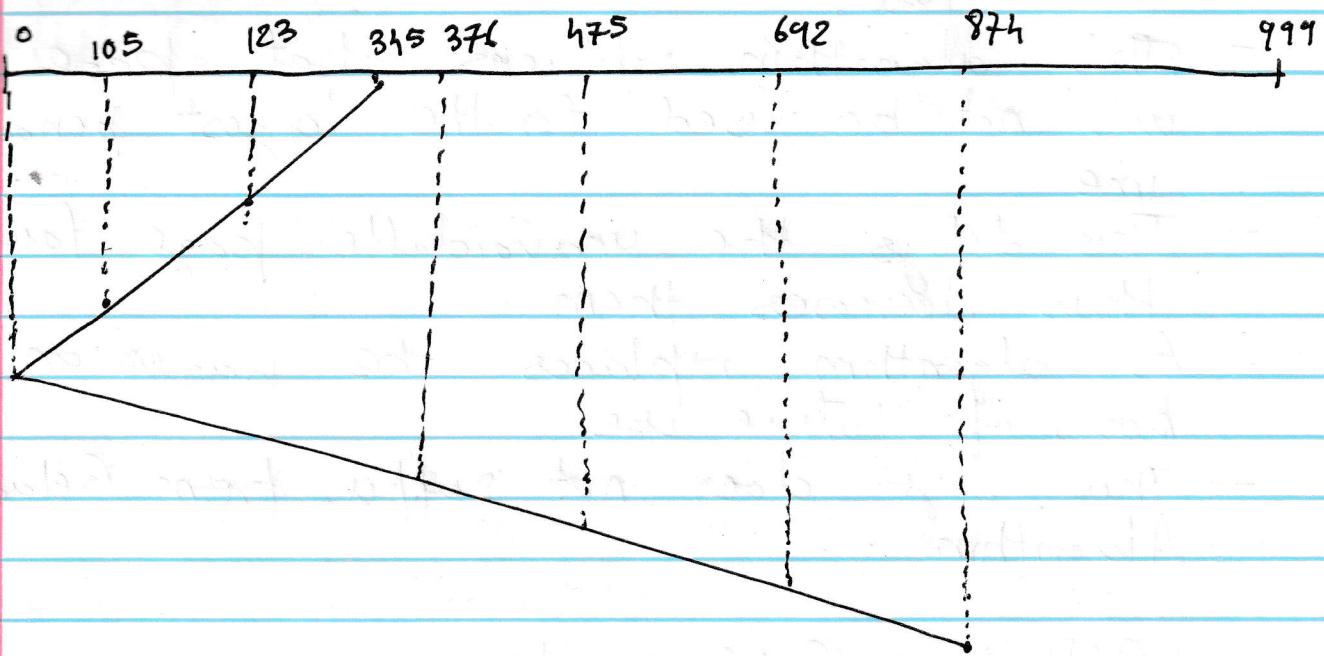
Total numbers of tracks  
Disk arm moved  
 $= (345 - 123) + (874 - 123)$   
 $+ (874 - 105) + (376 - 105)$   
 $= 2013$

ii) SSTF :



Total number of tracks disk arm moved  
 $= (874 - 345) + ((875) - (105))$   
 $= 1298$

iii) SCAN:



Total number of tracks moved by disk arm  
 $= (345 - 0) + (874 - 0)$   
 $= 1219$

Q3) FIFO Page Replacement:

- The simplest page-replacement algorithm & works on the basis of FIFO. It throws out the pages in the order in which they were brought in
- The time is associated with each page when it was brought into main memory
- This algorithm suffers from Belady's algorithm

Optimal Page Replacement

- This algorithm has the lowest page-fault rate
- This algorithm replaces that page which will not be used for the longest period of time
- This delays the unavoidable page faults thus decreases them
- As algorithm replaces the pages on the basis of future use
- This algo. does not suffer from Belady's Algorithm

### LRU Page Replacement:

- The time of pages last use is associated with each page.
- When a page must be replaced, LRU chooses that page that was used farthest back in the past.
- LRU is a good opposition as optimal page replacement.
- LRU looks backward in time while optimal page replacement looks forward in time.
- This algo. does not suffer through Belady's algorithm anomaly.

#### i) FIFO:

1	2	3	4	2	1	5	6	2	1	2	3
F	F	F	F	F	H	F	F	F	F	F	H

Daryl Fernandes 2020012004 (72)

5

7 3 7 1	6 3 7 G	3 3 7 G	2 2 X 6	1 2 1 6	2 2 1 X	3 2 1 3	6 1 1 3
F	F	H	F	F	H	F	F

Faults = 16

Hits = 4

ii) OPT:

1 1	2 2	3 2 8	4 2 4	2 2 4	1 2 4	5 2 5	6 2 6	2 2 6	1 1 6	2 2 6	3 2 6
F	F	F	F	H	H	F	F	H	H	H	F

7 3 2 7	6 3 2 6	3 3 2 6	2 3 2 6	1 3 2 1	2 3 2 1	3 3 2 1	6 3 2 6
F	F	H	H	F	H	H	F

Faults = 11

Hits = 9

iii) LRU:

1 1	2 2	3 2 3	4 2 3	2 2 3	1 2 1	5 2 1	6 5 1	2 5 1	1 5 1	2 5 1	3 2 2
F	F	F	F	H	F	F	F	F	F	F	F

7	6	3	2	1	2	3	6
3	3	3	3	3	3	3	3
7	6	6	6	1	1	1	6
2	2	2	2	2	2	2	2
F	F	H	H	F	H	H	F

Faults = 15

Hits = 5

Q4) File Allocation :

- The direct access nature of disks gives us flexibility in the implementation of files
- There should be optimum & effective utilization of disk space. The allocation of disk space to files should always allow for the optimum & effective utilization of disk space so that quick access to the files will be supposed.

① Contiguous Allocation :

- Each file takes up a set of contiguous blocks on the disks.
- Disk addresses define a linear ordering on the disk.
- Contiguous allocation is defined by the disk address of the first block & length of the file.
- The direction for each block requires specifies the address of the starting block & the total number of block allocated to this file.

### ② Linked list allocation:

- Each file is a linked list of disk blocks
- The scattered disk on the disk can be allocated to the file. The directory contains a pointer to the first & last blocks of the file.
- Creation of a new file is simple - new entry in the directory is created.
- The directory entry holds a pointer to the first disk block of the file.
- The pointer value is null for empty file & file size for this file is 0

### ③ Indexed allocation:

- In indexed allocation, all pointers are kept in one location called as index block.
- There is an index block assigned to each file & this index block holds the disk addresses of that particular file.
- There is a pointer from  $i^{th}$  entry in the index block to the  $i^{th}$  block of the file.
- It means  $n^{th}$  entry in index block holds the address of the  $n^{th}$  block of disk.
- The address of the index block of the file is maintained in the directory.