RAJESH DALAL

9049640271?

C-13

(Assignment-2)

Operating System

(a) Explain monitors and their uses in brief.
→ A monitor is a output device of the computer. It can be referred to as the visual interface b/w the user & the computer. It navigates us to what we are doing. It displays text, graphics, images, videos etc.
Various types of Monitors are:
→ CRT
→ LED
→ LCD
→ TFT
→ PLASMA
Main function of monitor is to accurately & clearly displays the programs, software or video being shown to user.

(b) Explain seek time and rotational latency in disk scheduling?
↳
Seek time → Time taken by (R-W) or Read-write head to reach the desired track from its current position
Rotational latency-time → Time taken by the sector to come under R-W head.

(c) What are various file attributes?
↳ These file attributes are
* NAME → It is the only information which is in human readable form.
* Identifier → The file is identified by a unique tag (number) within file system.

* Type → It is needed for systems that supports different types of files.
* Location → Pointer to file location on device.
* Size → current size of file
* Protection → This controls and assigns the power of reading, writing, executing

(d) Difference b|w Physical and logical file.

↳

Physical file	Logical file.
1) occupies portion of memory. It contains original data	1) doesn't occupy any memory space doesn't contain any data
2) A physical file contains one record format	2) It can contain up to 32 record formats
3) can exist even without LF	3) can't exist without PF
4) If there is a logical file for a PF, the PF can't be deleted until & unless we delete LF	4) If there is a Logical file for PF, LF can be deleted without deleting PF
5) CRTPF command is used to create such object	5) CRTLF command used to create such type object

(e) what is resource allocation graph and wait for graph?

Spural

↳ Resource allocation graph is the pictorial representation of the state of a system. As its name suggests, the resource allocation graph is the complete information about all the processes which are holding some resources or waiting some resources.

RAG contains vertex + edges

1) Process vertex → every process will be represented as a process vertex. Generally, process will be represented with circle.

2) Resource vertex → every resource will be represented as resource vertex. It is of 2 types → single instance
→ Multiple instance type resource.

Wait for graph → It is a directed graph used for deadlock detection in operating systems and relational database systems.

(8) A disk with 1000 cylinder numbered 0 to 999. Assume last request serviced was at track 344 and head is moving towards 0. The queue in order contain request like
122, 875, 682, 344, 455, 100, 390
Perform computation for following algo
(i) FIFO (ii) SSTF (3) SCAN
(4) C-SCAN (5) LOOK (6) C-LOOK

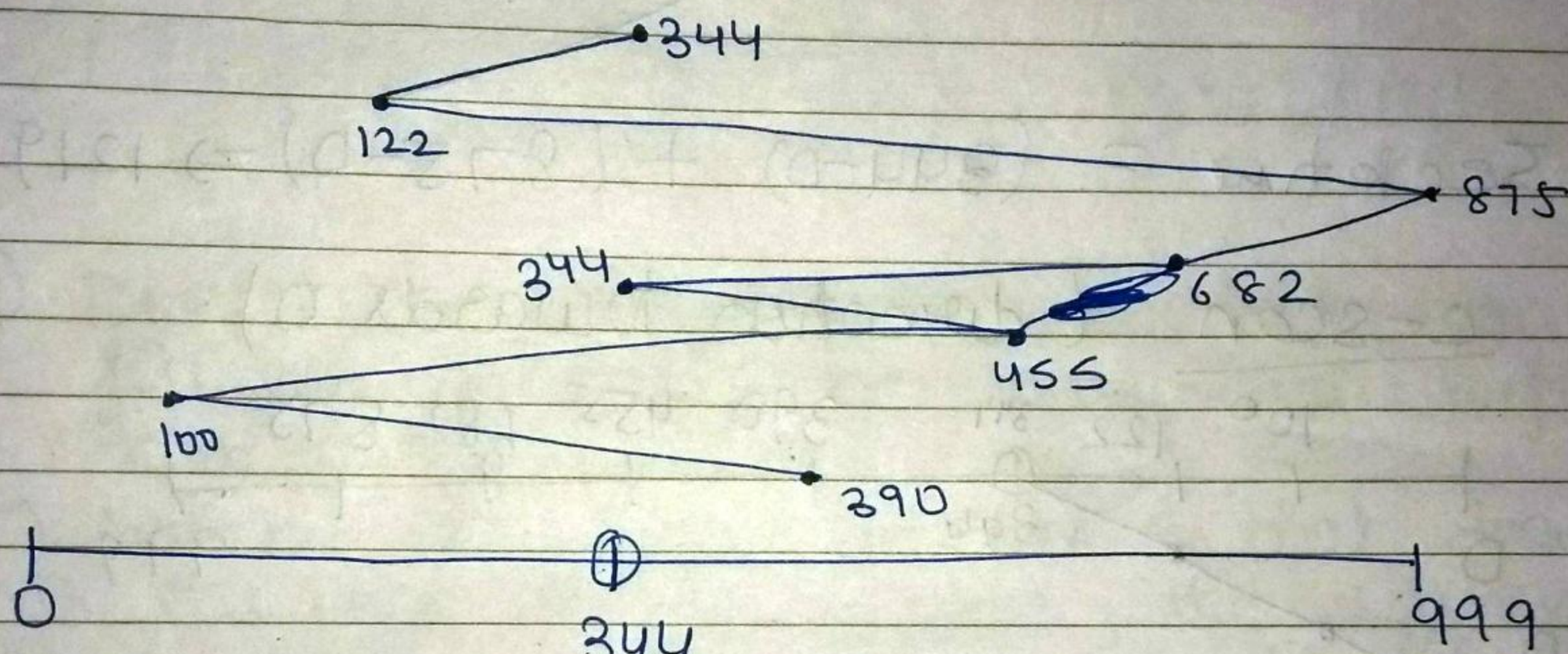
Spiral

① FIFO start (344)

Request (122, 875, 682, 344, 455, 100, 390)
Queues (0, 122, 875, 682, 344, 455, 100, 390, 999)

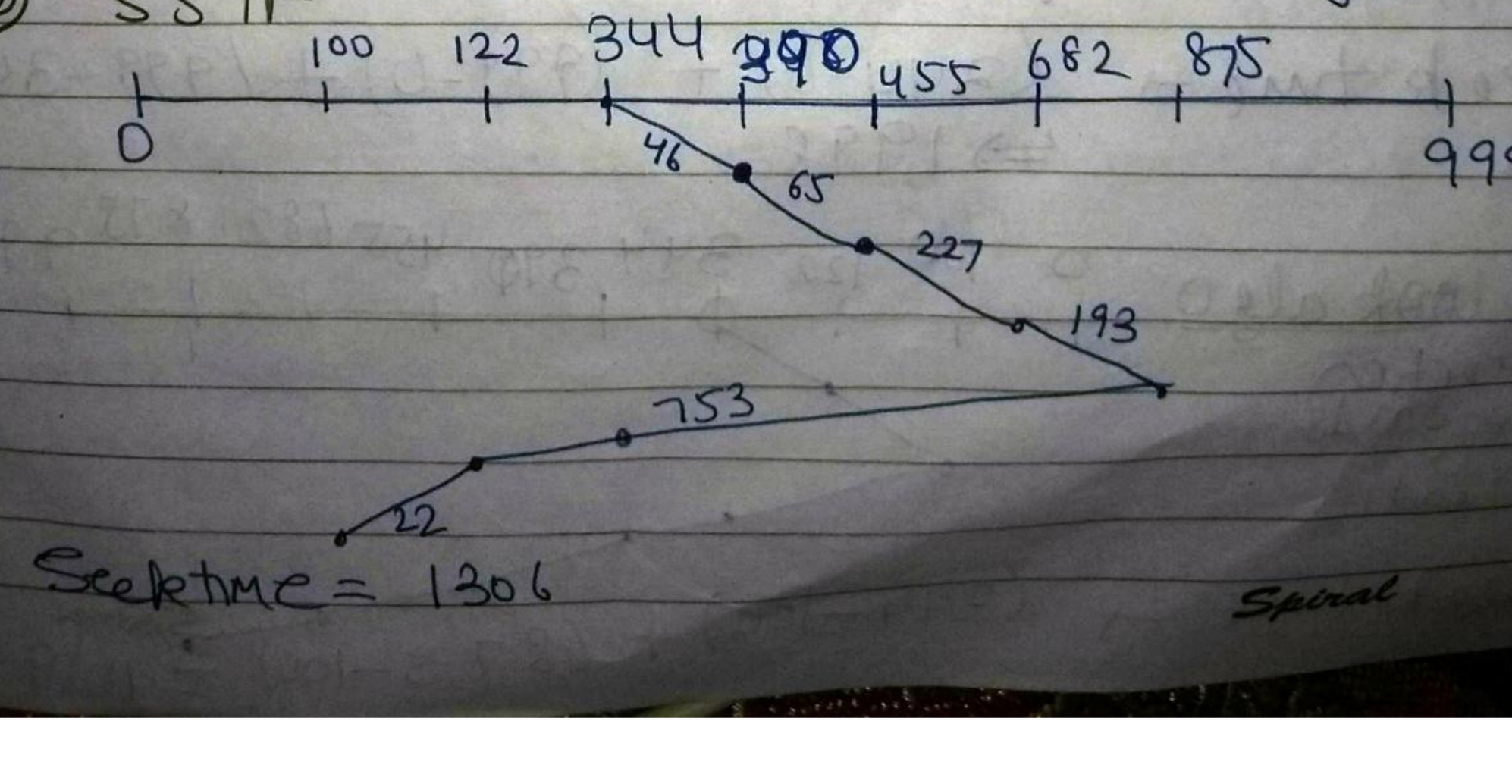
↳ Arrange in increasing

0 100 122 344 390 455 682 875 999



seek time = (344-122) + (875-122) + (682-8
+ (875-682) + (682-344)
+ (455-344) + (455-100) + (390-100)
= 2262

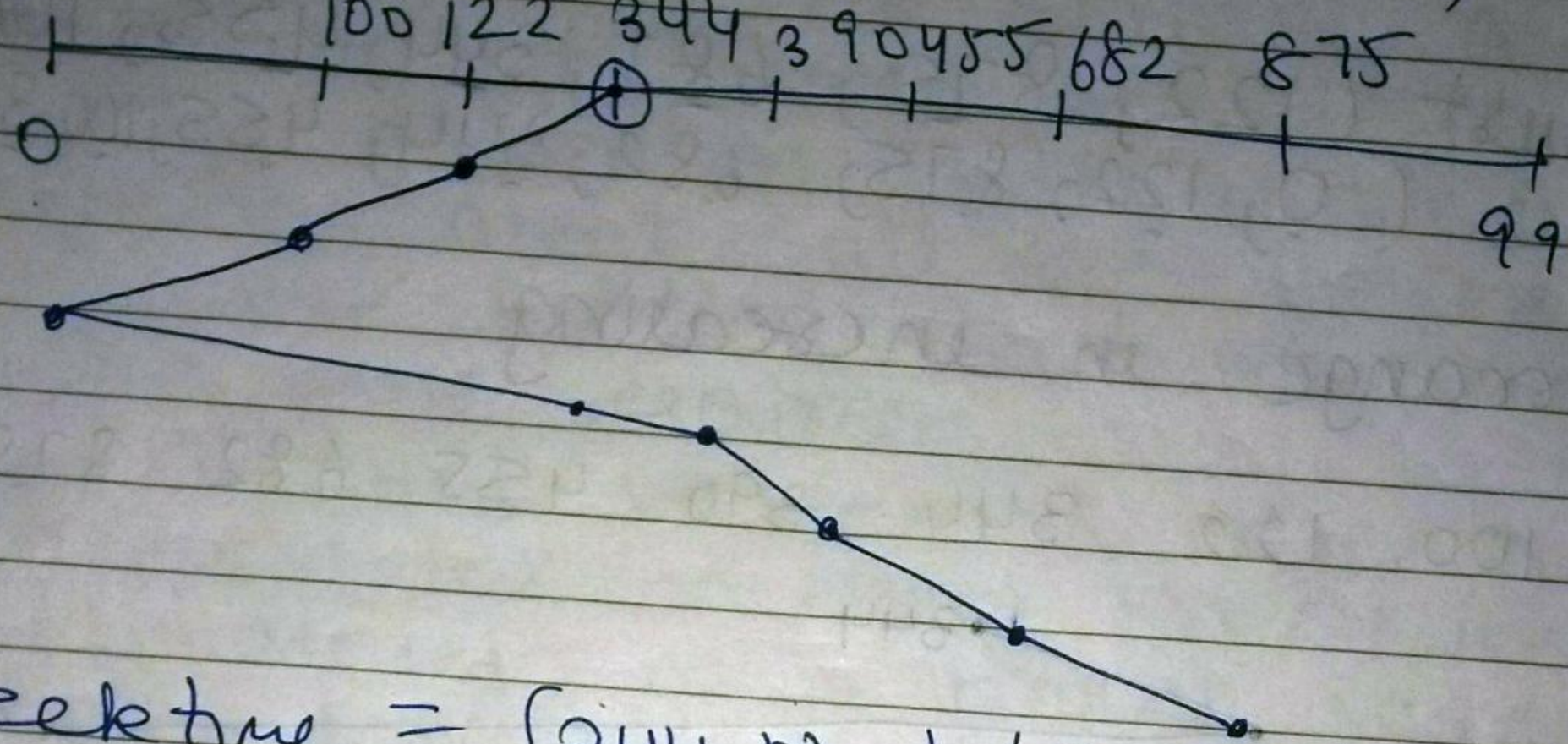
② SSTF → shortest seek time algo



Seek time = 1306

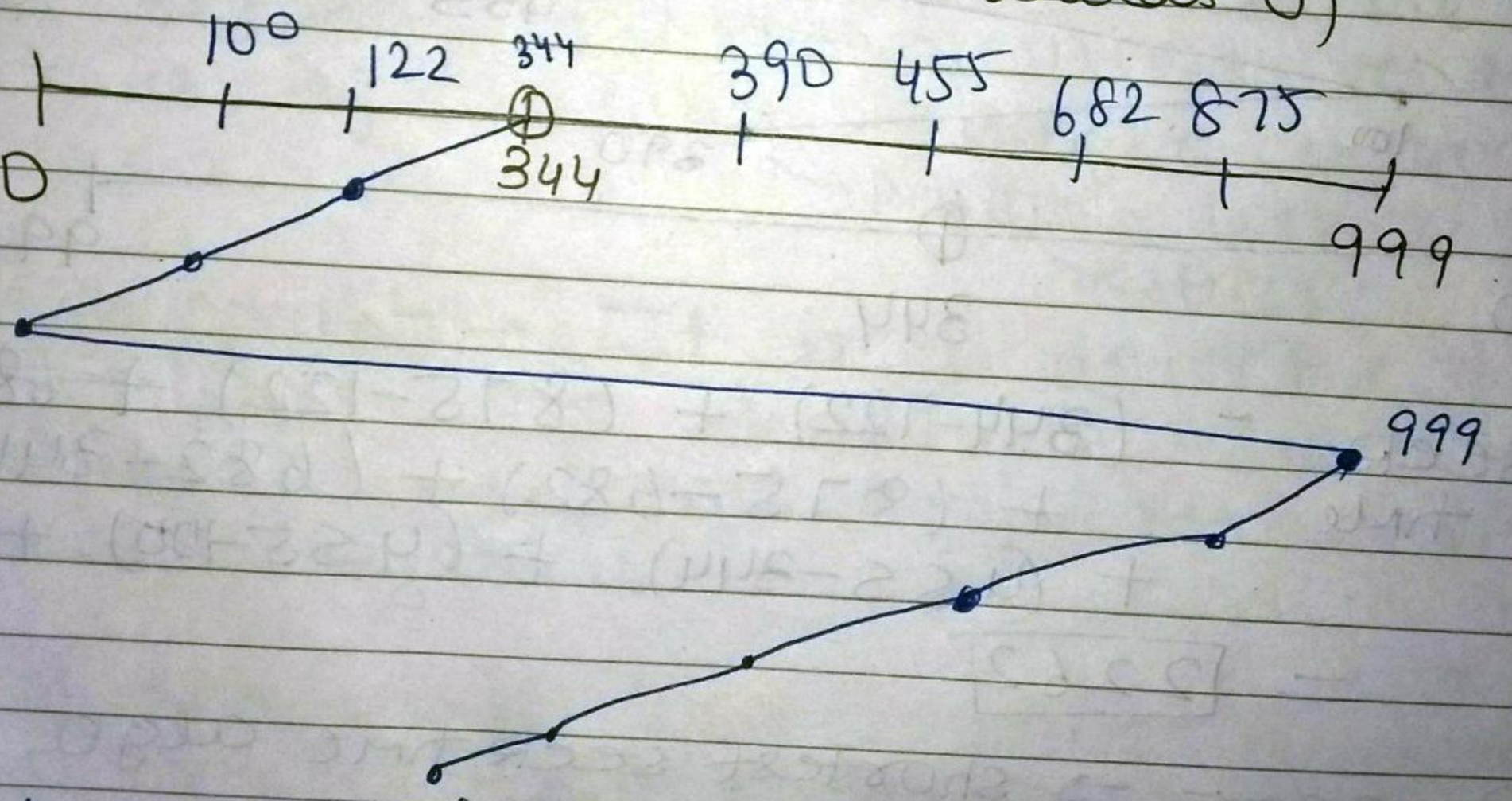
Spiral

③ SCAN (head towards 0)



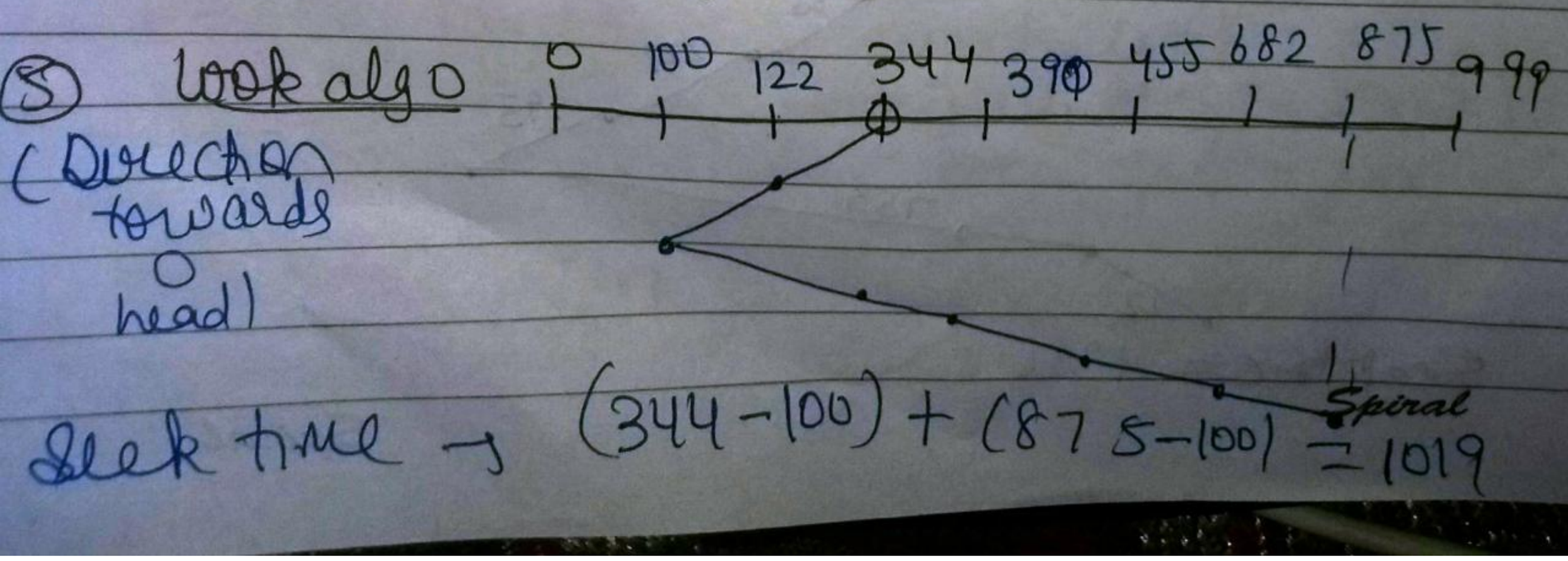
Seek time = (344-0) + (875-0) ⇒ 1219

④ C-scan (direction towards 0)



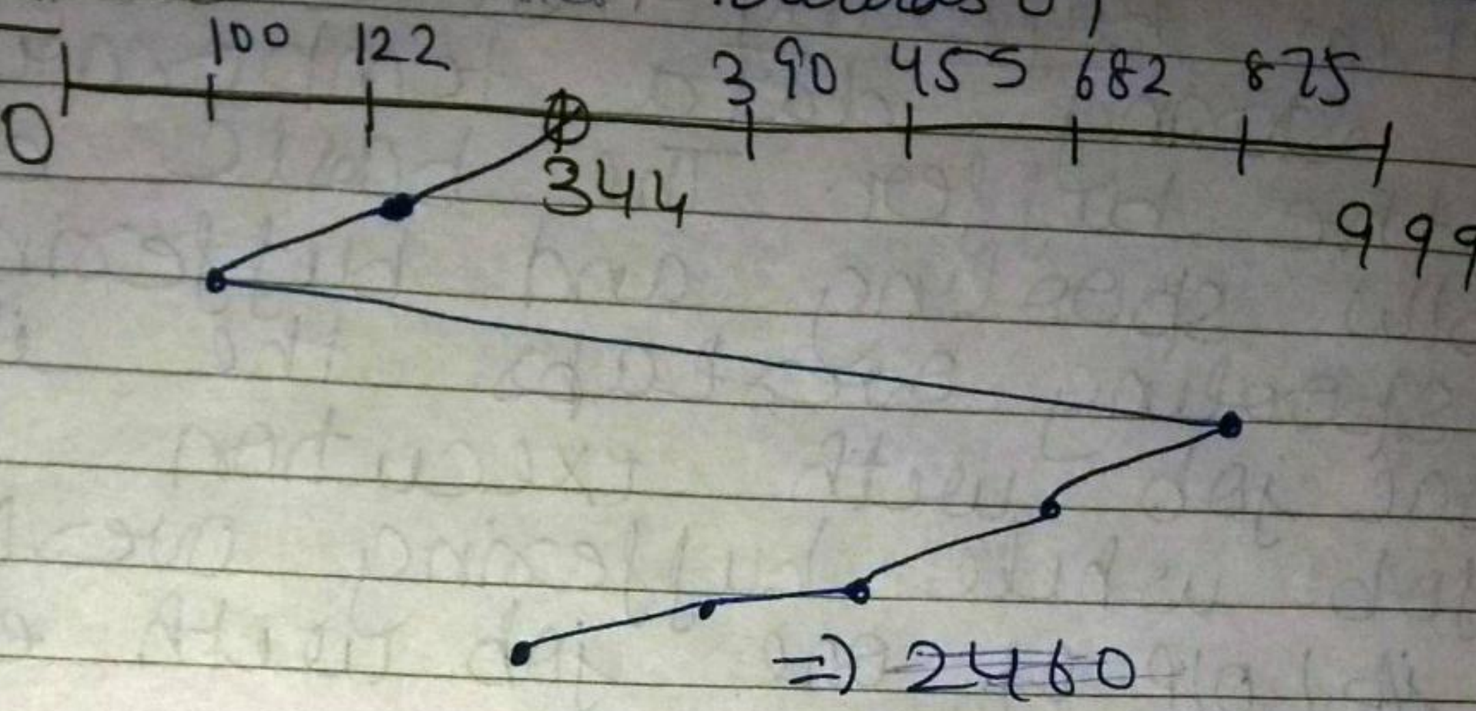
seek time → (344-0) + (999-0) + (999-344)
⇒ 1,998.

⑤ Look algo (Direction towards 0 head)



Seek time → (344-100) + (875-100) = 1019

(6) C-LOOK (direction towards 0)



Seek time = (344-100) + (875-100) + (875-344) = 1550

(9) During recovery from deadlock, what factors should be considered before terminating process?

→ When an algo determines that deadlock has occured in the system, the system must recover from that deadlock. There are two approaches of breaking a deadlock

(1) Process termination → to eliminate deadlock we can kill one or more processes.
   (a) Abort all deadlocked processes
   (b) Abort one process at a time until deadlock is eliminated.

(2) Resource Preemption :-
   (a) Selecting a victim
   (b) Rollback
   (c) Starvation.

(i) (a) Buffering → It is an act of storing data temporarily in the buffer. The basic difference blw spooling and buffering is that spooling overlaps the i/p / o/p of one job with execution of another job while buffering overlaps the i/p / o/p of one job with execution of same job.

(b) Multiplexing → It is a way of sending multiple signals or streams of information over a communication link at the same time in the form of a single, complex signal, the receiver recovers the separate signals, a process called demultiplexing.

(c) channel → It is an independent hardware component that coordinate I/O to a set of controllers. Computer systems that use I/O channel have special hardware components that handles all I/O operations. channels uses separate, independent & low cost processor for its functioning which are called channel processors.

(d) caching → Catche is a type of memory that is used to increase the speed of data accus. Normally data required for any process resides in the main memory. However, it is transferred to

Spiral

to cache memory temporarely if it is used frequent enough.

(1) Consider system with 6 resources of same type that are shared by 4 processes each Process need atmost 3 resource to execute shows that system is deadlock free.

↳ R=6 P_1 P_2 P_3 P_4

let check for 1 atmost req.

P_1	P_2	P_3	P_4	
1	1	1	1	→ get execute

No deadlock

↳ let req is 2 for execution

P_1	P_2	P_3	P_4
1	1	1	1
1	1		

P_1 + P_2 will execute and will make 4 resource free then P_3, P_4 execute so deadlock free

↳ for 3 atmost

case 1

P_1	P_2	P_3	P_4
1	1		
1	1		
1	1		

P_1 + P_2 execute

Similarly P_3 + P_4

case 2

P_1	P_2	P_3	P_4
1	1	1	1
1			
1			
1			

P_1 execute 3 get free then P_2 & so on

case 3

P_1	P_2	P_3	P_4
1	1	1	
1	1		
1			

P_1 execute then P_2, P_3, P_4

Hence in all cases there is no deadlock

(R)

	Allocation A B C D	MAX A B C D	Available A B C D
P_0	0 0 2 3	0 0 2 3	1 5 2 0
P_1	0 0 0 0	1 7 5 0	
P_2	2 4 5 4	3 4 5 6	
P_3	0 5 4 2	0 5 6 2	
P_4	0 1 2 5	0 7 6 7	

Need Matrix

	A	B	C	D
P_0	0	0	0	0
P_1	1	7	5	0
P_2	1	0	0	2
P_3	0	0	2	0
P_4	0	6	4	2

Safety algo → If (need ≤ available)
then
execute Process
[new ava. = ava. + allocation]
else
{don't execute go forward.

algo

↳ check Need P_0 ≤ available 1 5 2 0
so P_0 execute (P_0)
new av. = 0 0 0 0 + 1 5 2 0 = 1 5 2 0
↳ check P_1 1 7 5 0 with 1 5 2 0
↳ check P_2 1 0 0 2 with 1 5 2 0
↳ check P_3 0 0 2 0 with 1 5 2 0
So execute (P_3)
new avail. = 0 1 5 4 0
↳ check P_4 0 6 4 2 with 1 5 4 0
↳ check P_1 1 7 5 0 with 1 5 4 0

Spiral

Apply algo

↳ check Need of $P_0 \leq$ 1520
So execute (P_0)
New ava. = 1543
↳ check P_1 (1750) with new ava.
↳ check P_2 (1002) with (1543)
(P_2) execute (new av. = 3997)
↳ check P_3 (0020) with 3997
(P_3) execute (new av = 3 14 13 9)
↳ check P_4 0642 with (3,14,13,9)
(P_4) execute
↳ check P_1 (1,7,5,0) with (3,14,13,9)
(P_1) execute

Safe sequence is

$P_0 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_1$

If request P_1 arrives (0420) can it be granted?

↳

Resource request algo

(1) If request ≤ need, go to step 2
else error
(2) If request ≤ available, go to step 3
else wait
(3) Ava. = Ava. − Request
Allocation = Allocation + request
Need = Need − Request
(4) check new state is safe or not.

→ Request (0 4 2 0) P_1
Need P_1 (1 7 5 0)
(1) Check request ≤ need (true)
(2) check (0 4 2 0) ≤ (1 5 2 0) true

Spural

Available = 1 5 2 0
− 0 4 2 0
1 1 0 0

New available → 1100

Allocation = 0 0 0 0
0 4 2 0
New allocation → 0 4 2 0

Need = 1 7 5 0
0 4 2 0
1 3 3 0

Now Make new table

	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P_0	0	0	2	3	0	0	2	3	1	1	0	0
P_1	0	4	2	0	1	7	5	0				
P_2	2	4	5	4	3	4	5	6				
P_3	0	5	4	2	0	5	6	2				
P_4	0	1	2	5	0	7	6	7				

Need Matrix

	A	B	C	D
P_0	0	0	0	0
P_1	1	3	3	0
P_2	1	0	0	2
P_3	0	0	2	0
P_4	0	6	4	2

Spiral

#Now check whether safe or not?
Ans. ① check P_0 need $\leq$ Available
(P_0) execute
new AV. = 1123
② check P_1 (1 3 3 0) $\leq$ (1 1 2 3)
③ check P_2 (1 0 0 2) $\leq$ (1 1 2 3)
(P_2) execute
new AV. → 3 5 7 7
④ check P_3 (0 4 2 0) $\leq$ 3 5 7 7
(P_3) execute
New AV. = 3 10 11 9
⑤ check P_4 (0 6 4 2) $\leq$ (3 10 11 9)
P_4 execute
⑥ execute P_1

#Sequence of executing → $P_0 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_1$

Explain free space management?
↳ The system keeps tracks of the free disk blocks for allocating space to files when they are created. Also to reuse the space released deleting the files, free space management becomes crucial.