Kulvir Singh
19BCE2074

DATA STRUCTURES DA-1 (Theory)

8) The data structure which suits the application is QUEUES.
A circular queue would be implemented for the application.

Algorithm.

- Create a queue with a particular max size
- The Queue can perform these 2 operations.
  (i) Enqueue (ii) Dequeue.

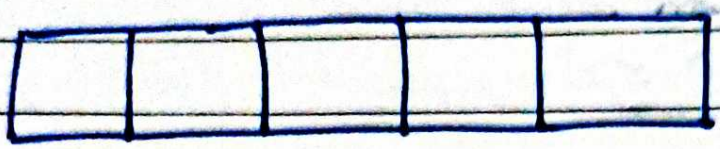
(i) Enqueue. – for insertion of jobs.
1. check if the rear pointer is pointing at the last position of queue
2. if yes then do not insert a job.
3. if no then insert the job.

(ii) Dequeue – for completion of 1st job.
1. check for an empty queue by checking the front pointer pointing to a job.
2. if yes then remove the job by increasing the front pointer
3. if no then display an error message

Diagramatic Representation :

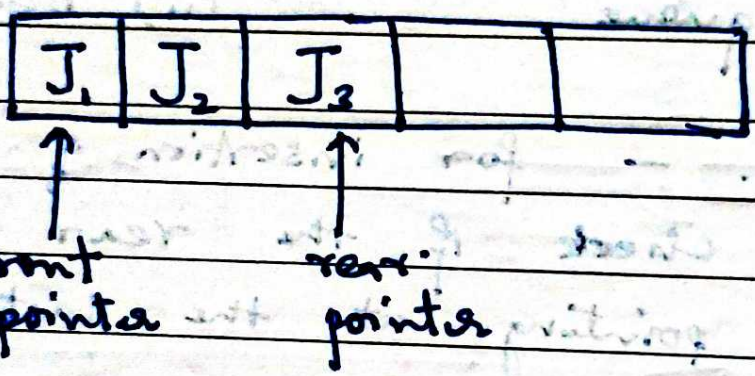
(1) Creation of a queue.

Eg:- The queue can store 5 jobs at max at a time.

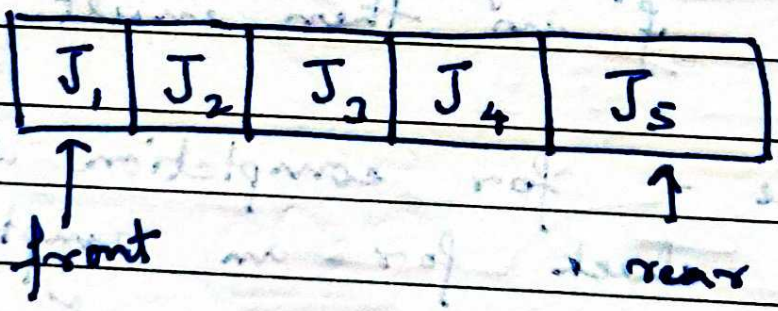


(2) Enqueue ⇒ inserting jobs.

Eg:- The user enters 3 jobs at a time.



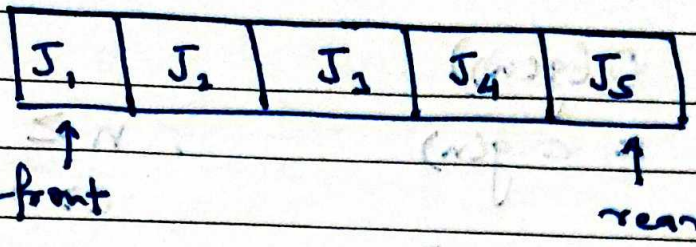
If the user enters another 2 jobs



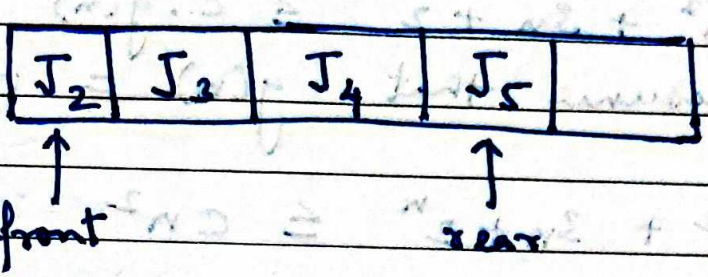
Now no more jobs can be entered as the queue is full.

(3) Dequeue ⇒ Completion of 1st job.

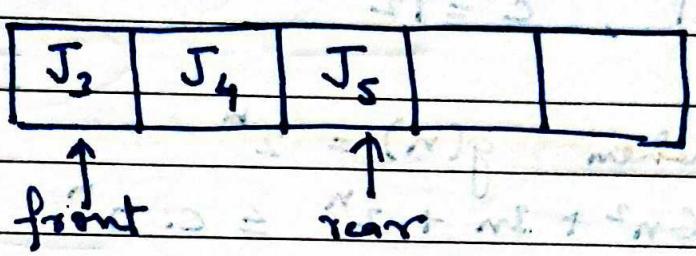
Initially:



1st Job complete (i):



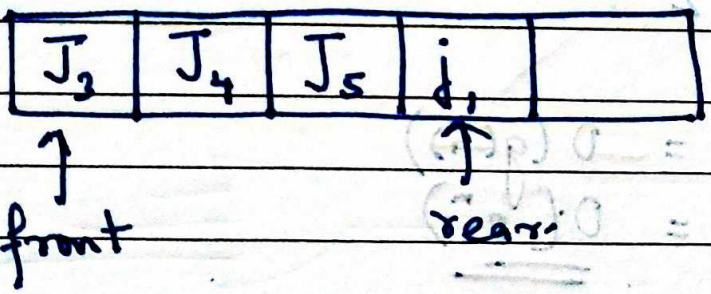
1st Job complete (ii):



On completion of jobs an empty space is created and hence more jobs can be added

Eg:

Adding another job j_1



8) (a) $f(n) = 6n^2 + 3n + 2^n$

$f(n) = O(g(n))$

$\therefore f(n) \leq c\, g(n)$, $n \geq n_0$

$c > 0$, $n_0 \geq 1$

$\Rightarrow 6n^2 + 3n + 2^n \leq c \cdot g(n)$

Let us assume that, $g(n) = n^2$

$\Rightarrow 6n^2 + 3n + 2^n \leq cn^2$

when $n = 1$, $c = 12$

However when $g(n) = 2^n$

$\Rightarrow 6n^2 + 3n + 2^n \leq c \cdot 2^n$

when $n = 1$, $c = 6$.

$\therefore$ 2^n is more dominant when we consider higher values of n such as 1000 , 10^5 etc.

$\therefore$

$g(n) = 2^n$

$f(n) = O(g(n))$

$= O(2^n)$

8. (b) is $5n^2 + 8n + 12 = O(n^3)$?

If $f(n) = O(g(n))$,
then $f(n) \leq c \cdot g(n)$. $n \geq n_0$
$c > 0$, $n_0 \geq 1$

$\Rightarrow 5n^2 + 8n + 12 \leq c \cdot n^3$

$\Rightarrow cn^3 - 5n^2 - 8n - 12 \leq 0$.

for any value of c., the roots of the equation are imaginary.

for instance, if $c = 1$, then

$$n^3 - 5n^2 - 8n - 12 \leq 0$$

$$n \leq 6.511, \; -0.75 + i, \; -0.15 + i$$

which is not possible as $n \geq n_0 \geq 1$

Hence $f(n) = 5n^2 + 8n + 12 \neq O(n^3)$.

11) $f(n) = 3 * 2^n + 4n^2 + 5n + 2$

$\theta(g(n)) = f(n)$

Such that

$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ $\in n \geq n_0$

$c_1(g(n)) \leq 3 * 2^n + 4n^2 + 5n + 2 \leq c_2 g(n)$

$\therefore$ let us assume $g(n)$ to be 2^n

$c_1 * 2^n \leq 3 * 2^n + 4n^2 + 5n + 2 \leq c_2 * 2^n$

$\therefore$ there are c_1 and c_2 which satisfies the above condition for $n \geq$ some n_0.

$\therefore$ Theta notation for $f(n)$ is

$\theta(2^n)$.

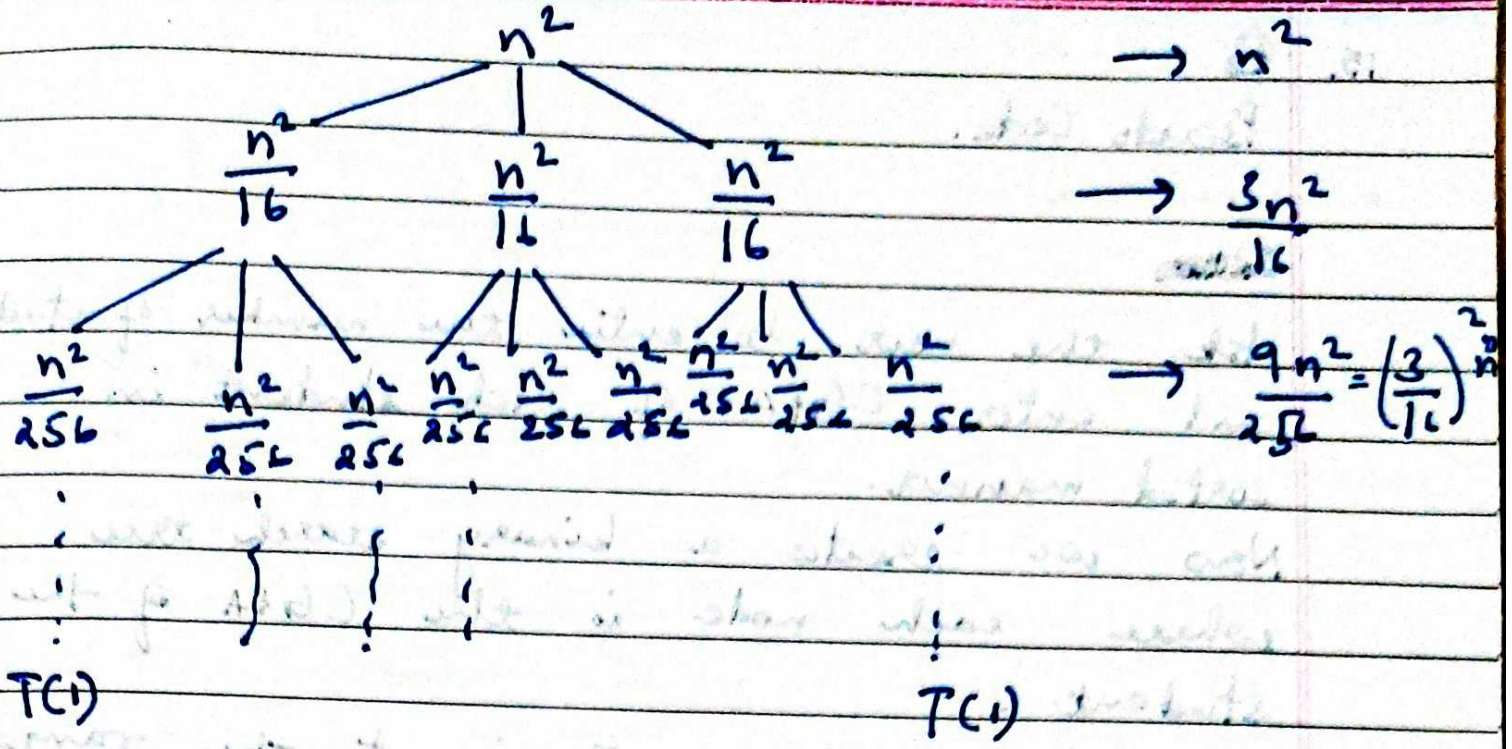
13) $T(n) = 3T(n/4) + \Theta(n^2)$

$T(n) = 3T(n/4) + n^2$

$T(n/4) = 3T(n/16) + (n/4)^2$

$T(n/16) = 3T(n/64) + (n/16)^2$

$T(n/64) = 3T(n/256) + (n/64)^2$



Depth $\Rightarrow \frac{n}{b^i} = 1$

$\Rightarrow \frac{n}{4^i} = 1$

$\Rightarrow n = 4^i$

$\Rightarrow \log n = i \log 4$

$\Rightarrow \boxed{i = \log_4 n}$

$$T(n) = n^2 + \frac{3}{16}n^2 + \left(\frac{3}{16}\right)^2 n^2 + \cdots + \left(\frac{3}{16}\right)^{i-1} n^2$$

$$= n^2 \left(1 + \frac{3}{16} + \left(\frac{3}{16}\right)^2 + \cdots + \left(\frac{3}{16}\right)^{i-1}\right)$$

$$\boxed{T(n) = \theta(n^2)}$$

15. (a)

Pseudo Code.

Ask the user to enter the number of students and enter CGPA of each student in sorted manner.

Now we create a binary search tree where each node is the CGPA of the student.

We ask the user to input the range that the user requires.

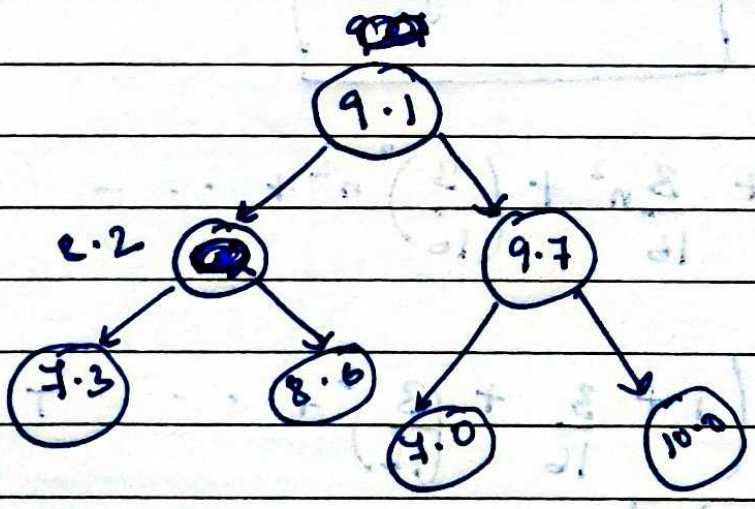
For eg. →

No. of students = 7

CGPA = 7, 7.3, 8.2, 8.6, 9.1, 9.7, 10

Range = 8.5 to 10.0

Now,



After creating the tree we travel in the tree by Inorder traversal.
The inorder traversal will return elements and we will store it in an array.
before storing, we will apply the condition that the CGPA should be in the range of $8.5 \leq CGPA \leq 10.0$
so the array would be

8.6	9.1	9.7	10.0

Print the array for required output.