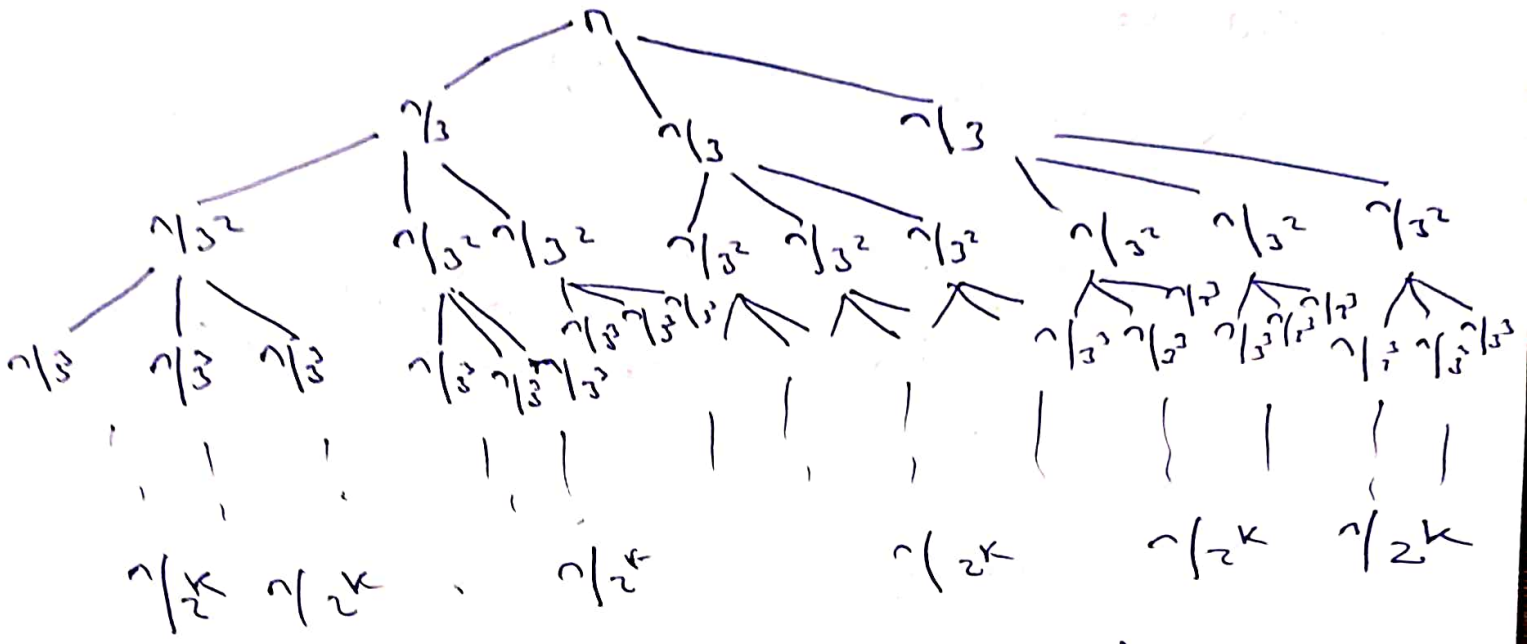
Part B Set 8

DSA

YASH GUPTA
S2020010219

(6)

$T(n) = 3\,T(n/3) + n$



Total steps $= k$ $\quad k = \log n$

Total time $= n \log n$

Hence $O(\log n)$

→ $T(n) = aT(n/b) + cn^k$

$T(n) = 3\,T(n/3) + n$

$a = 3$, $b = 3$, $k = 1$, $c = 1$

$b^k = 3$

$a = b^k$

$T(n) = \theta(n^k \log n)$

$= \theta(n \log n)$

YASH GUPTA
S20200010239

1 (a)

-> find the smaller of two numbers

-> if smaller divides larger
↳ then GCD = smaller number

-> else if for(int i = smaller/2 ; i >= 2 ; i--)
{ if i divides both numbers
↳ then GCD = i
}

-> else return 1

Computational complexity = $O(n)$

Q(a)

→ Compare the key with middle element.

- if it matches
  return index
- else if key > mid element
  key can only be right side
  Set mid point now to the right half
- else if key < mid element
  key can only be left side
  Set mid point now to the left half

Pseudocode

arr → sorted array of size n
key → value to search

```
Set   low = 1
Set   up = n
While key not found
      Set mid = low + (up - low)/2
      If  arr[mid] < key
          Set low = mid + 1
      If  arr[mid] > key
          Set up = mid - 1
      If  arr[mid] = key
          return mid (location)
End while
```

continue →

YASH GUPTA
5202001234

Time complexity = $O(\log n)$

Space complexity = $O(n)$

2(b)

→ It saves time when we have to go to first node from last node.

→ It helps in the implementation of queue

→ In circular list you can go to previous node.

For eg Circular list is ~~not~~ used in game play, it gives turn to each player without any failure.

YASH GUPTA
S20200010234

2(c)

```
#include <stdio.h>
struct node { struct node * next;
};
```

typedef struct node;

```
void delete_atstart(**head)
{
    struct node *p;
    if (head == NULL)
    {
        printf("Underflow");
    }
    else if (head -> next == head)
    {
        head = NULL;
        free(head);
    }
    else
    {
        p = head;
        while (p->next != head)
        {
            p = p -> next;
        }
        p -> next = head -> next;
        p = head;
        head = head -> next;
        free(p);
    }
}
```

node *head;

YASH GUPTA
5720001231

```
int main()
{   struct node* head1;
    delete_start(head1);
    return
}
```

Assuming head1 is the circular linked list.

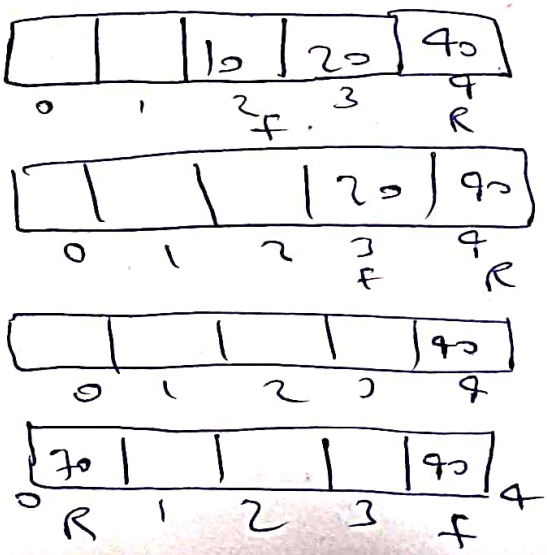
Q-3

(i) After inserting

$f = 2, \quad R = 1$

(ii) If we insert 30, R will become 2, then overflow.
Therefore f, R values will not change.

(iii)



YASH GUPTA
5220001029