Sneha rao

Tasks given for week 1:

1.Time Complexity
2.STL
3.Sorting
4.15 extra random questions

Time-Complexity:
Watched videos recommended and made notes for future reference

Stl:
Went through all known concepts and learnt few new ones as well and implemented these concepts in the 15 extra questions for practice.

Sorting:
Leet code: https://leetcode.com/problems/sort-an-array/
Bubble sort: (this question using bubble sort didnt satisfy all test cases due to execution time constraint)

Code:

```
class Solution {
public:
   vector<int> sortArray(vector<int>& nums) {
      int i,j;
      for(i=0;i<nums.size()-1;i++){
         for(j=0;j<nums.size()-i-1;j++)
            if(nums[j]>nums[j+1])
               swap(&nums[j],&nums[j+1]);

      }

      return nums;

   }

   void swap(int *xp,int *yp){
      int temp = *xp;
      *xp = *yp;
      *yp = temp;
```

```
    }
};
```

All other sorts i.e. bubble sort, bubble optimized sort, merge sort, heap sort, counting sort, quick sort, insertion sort, selection sort was coded on vscode and implemented after referring geeks for geeks links given in the DSA pdf.

Extra 15 questions for the week :

1.https://www.hackerrank.com/challenges/a-very-big-sum
Status: All test cases accepted
Code:

```
#include <bits/stdc++.h>

using namespace std;

vector<string> split_string(string);

// Complete the aVeryBigSum function below.
long aVeryBigSum(vector<long> ar) {

    long sum =0;
    for(int i=0;i<ar.size();i++){
        sum+=ar[i];
    }

   return sum;

}
```

2.https://www.hackerrank.com/challenges/mini-max-sum
Status:All test cases accepted
Code:

```
#include <bits/stdc++.h>
using namespace std;
vector<string> split_string(string);
// Complete the miniMaxSum function below.
void miniMaxSum(vector<int> arr) {

    vector<long> sum={0,0,0,0,0};
```

```
   int j=0;
   for(int i=0;i<arr.size();i++){
      for(j=0;j<arr.size();j++){
         if(j==i)
         continue;
         sum[i]+=arr[j];
      }

   }


  long max=sum[0];

  for(int i=0;i<sum.size();i++){
     if(sum[i]>max)
     max=sum[i];
  }

  long min = sum[0];

  for(int i=0;i<sum.size();i++){
     if(sum[i]<min)
     min=sum[i];
  }
  cout<<min<<"\t"<<max;
}
```

3.https://www.hackerrank.com/challenges/compare-the-triplets
Status: All test cases accepted
Code:

```
#include <bits/stdc++.h>

using namespace std;

string ltrim(const string &);
string rtrim(const string &);
vector<string> split(const string &);

// Complete the compareTriplets function below.
vector<int> compareTriplets(vector<int> a, vector<int> b) {
   vector<int> q={0,0};

   for(int i=0;i<3;i++){

      if(a[i]>b[i])
```

```
        q[0]=q[0]+1;

        if(a[i]<b[i])
        q[1]=q[1]+1;


    }
  return q;
}
```

4.https://www.hackerrank.com/challenges/plus-minus
Status:All test cases accepted
Code:

```
#include <bits/stdc++.h>
using namespace std;
vector<string> split_string(string);

// Complete the plusMinus function below.
void plusMinus(vector<int> arr) {
    float p=0;
    float n=0;
    float z=0;

    for(int i=0;i<arr.size();i++){
        if(arr[i]>0)
        p++;
        if(arr[i]<0)
        n++;
        if(arr[i]==0)
        z++;
    }

    cout<<std::fixed;
    cout<<std::setprecision(6);
    cout<<p/arr.size()<<"\n"<<n/arr.size()<<"\n"<<z/arr.size()<<"\n";
}
```

5.https://www.hackerrank.com/challenges/simple-array-sum
Status: All test cases accepted
Code:

```
#include <bits/stdc++.h>
using namespace std;
vector<string> split_string(string);
/*
```

```
 * Complete the simpleArraySum function below.
 */
int simpleArraySum(vector<int> ar) {
    /*
     * Write your code here.
     */int sum=0;
     for(int i=0;i<ar.size();i++){
       sum+=ar[i] ;
     }
return sum;
}
```

6.https://www.hackerrank.com/challenges/migratory-birds

Status: one test case failed all other test cases successful(test case 2 failed)

Code:

```
#include <bits/stdc++.h>
using namespace std;

string ltrim(const string &);
string rtrim(const string &);
vector<string> split(const string &);

// Complete the migratoryBirds function below.
int migratoryBirds(vector<int> arr) {

    int max = *max_element(arr.begin(),arr.end());
    int min = *min_element(arr.begin(),arr.end());
    int range = max-min+1;

    vector<long> count(range);
    for(int i=0;arr[i];i++)
    ++count[arr[i]];


        long m = count[0];
        int r;

        for(int i=0;i<range;i++){
            if(count[i]>m){
            m=count[i];
            r=i;

            }
        }

        return r;
```

```
    }
```

7.https://www.hackerrank.com/challenges/divisible-sum-pairs
Status: all test cases accepted
Code:

```
#include <bits/stdc++.h>

using namespace std;

vector<string> split_string(string);

// Complete the divisibleSumPairs function below.
int divisibleSumPairs(int n, int k, vector<int> ar) {

   int i,j;
   int count =0;
   for(i=0;i<n;i++){
      for(j=i+1;j<n;j++){
         if((ar[i]+ar[j])%k == 0){
            count++;
         }
      }
   }

   return count;
}
```

8.https://www.hackerrank.com/challenges/strong-password
Status: All test cases accepted
Code:

```
#include <bits/stdc++.h>
using namespace std;

// Complete the minimumNumber function below.
int minimumNumber(int n, string password) {
   // Return the minimum number of characters to make the password strong
   int i,p;
   int digit=0;
   int l,u,s;
   l=0;
   u=0;
   s=0;
   for(i=0;i<n;i++){
      if(isdigit(password[i]))
      digit++;
```

```
      if(islower(password[i]))
      l++;
      if(isupper(password[i]))
      u++;

if(password[i]=='!'||password[i]=='@'||password[i]=='#'||password[i]=='$'||password[i]=='%'|
|password[i]=='^'||password[i]=='&'||password[i]=='*'||password[i]=='-'||password[i]=='+'||p
assword[i]=='('||password[i]==')')
      s++;

   }
 if(n<6&&digit!=0&&l!=0&&u!=0&&s!=0)
    p = 6-n;

 if(digit==0&&l!=0&&u!=0&&s!=0){
 if(n<6)
    p = 6-n;
 if(n>=6){
  p = 1;
 }

 }

 if(digit!=0&&l==0&&u!=0&&s!=0)
 {
 if(n<6)
    p = 6-n;
 if(n>=6)
    p = 1;

 }


 if(digit!=0&&l!=0&&u==0&&s!=0)
 {
 if(n<6)
    p = 6-n;
 if(n>=6)
    p = 1;

 }


 if(digit!=0&&l!=0&&u!=0&&s==0)
 {
```

```
if(n<6)
   p = 6-n;
if(n>=6)
   p = 1;

}


if(digit==0&&l==0&&u!=0&&s!=0){

if(n<6){
   if(6-n>=2)
    p = 6-n;

   else if(6-n<2)
    p = 2;
 }
if(n>=6)
    p = 2;
}

if(digit==0&&l!=0&&u==0&&s!=0)
{

if(n<6){
   if(6-n>=2)
    p = 6-n;

   else if(6-n<2)
    p = 2;
 }
if(n>=6)
    p = 2;
}

if(digit==0&&l!=0&&u!=0&&s==0)
{

if(n<6){
   if(6-n>=2)
    p = 6-n;

   else if(6-n<2)
    p = 2;
 }
```

```
if(n>=6)
    p = 2;
}

if(digit!=0&&l==0&&u==0&&s!=0)
{

if(n<6){
   if(6-n>=2)
    p = 6-n;

   else if(6-n<2)
    p = 2;
 }
if(n>=6)
    p = 2;
}

 if(digit!=0&&l==0&&u!=0&&s==0)
{

if(n<6){
   if(6-n>=2)
    p = 6-n;

   else if(6-n<2)
    p = 2;
 }
if(n>=6)
    p = 2;
}

if(digit!=0&&l!=0&&u==0&&s==0)
{

if(n<6){
   if(6-n>=2)
    p = 6-n;

   else if(6-n<2)
    p = 2;
 }
if(n>=6)
    p = 2;
}
```

```
if(digit==0&&l==0&&u==0&&s!=0){
   if(n<6){
      if(6-n>=3)
       p = 6-n;

      else if(6-n<3)
          p = 3;

   }
   if(n>=6)
     p = 3;
}

if(digit==0&&l!=0&&u==0&&s==0){
   if(n<6){
      if(6-n>=3)
       p = 6-n;

      else if(6-n<3)
          p = 3;

   }
   if(n>=6)
    p = 3;
}

if(digit==0&&l==0&&u!=0&&s==0){
   if(n<6){
      if(6-n>=3)
      p = 6-n;

      else if(6-n<3)
          p = 3;

   }
   if(n>=6)
     p = 3;
}

if(digit!=0&&l==0&&u==0&&s==0){
   if(n<6){
      if(6-n>=3)
       p = 6-n;
```

```
        else if(6-n<3)
            p =3;

    }
    if(n>=6)
      p = 3;
 }

 if(digit==0&&l==0&&u==0&&s==0){
    if(n<6){
       if(6-n>=4)
       p = 6-n;

       else if(6-n<4)
           p = 4-(6-n);

    }
    if(n>=6)
      p = 4;
 }

 if(digit!=0&&l!=0&&u!=0&&s!=0&&n>6)
 p=0;

 return p;
 }
```

9.https://www.hackerrank.com/challenges/camelcase
Status: All test cases accepted
Code:

```
#include <bits/stdc++.h>
using namespace std;
// Complete the camelcase function below.
int camelcase(string s) {
   int count =0;
   int i;
   for(i=0;i<s.length();i++){
      if(isupper(s[i]))
      count++;
   }
 return count+1;
}
```

10.https://www.hackerrank.com/challenges/crush
Status: time limit exceeded for few test cases , all other test cases code success

Code:

```
#include <bits/stdc++.h>
using namespace std;
vector<string> split_string(string);
// Complete the arrayManipulation function below.
long arrayManipulation(int n, vector<vector<int>> queries) {
   vector<long> a(n,0);

   long max = a[0];

   for(int i =0; i< queries.size();i++){
      for(int j=queries[i][0]-1;j<queries[i][1];j++){
      a[j]+=queries[i][2];


   }
}

   for(int i= 0;i<n;i++){
      if(a[i]>max)
      max=a[i];
   }

      return max;
}
```

11.https://www.hackerrank.com/challenges/print-the-elements-of-a-linked-list
Status: All test cases accepted
Code:

```
void printLinkedList(SinglyLinkedListNode* head) {
   SinglyLinkedListNode* temp;
   temp = head;
   while(temp!=NULL){
      cout<<temp->data<<"\n";
      temp = temp->next;
   }
  }
```

12. https://www.hackerrank.com/challenges/diagonal-difference
Status: All test cases accepted
Code:

```
int diagonalDifference(vector<vector<int>> arr) {
   int i,j;
   int sump=0;
   int sums=0;
```

```
   int final =0;

   for(i=0;i<arr.size();i++){
      for(j=0;j<arr.size();j++){
         if(i==j)
         sump+=arr[i][j];

      }
   }

   int n= arr.size();

    for(i=0;i<arr.size();i++){
      for(j=0;j<arr.size();j++){
         if(i+j==n-1)
         sums+=arr[i][j];

      }
   }
  final = abs(sump-sums);
   return final;
}
```

13.https://www.hackerrank.com/challenges/staircase
Status:All test cases accepted
Code:

```
#include <bits/stdc++.h>

using namespace std;

// Complete the staircase function below.
void staircase(int n) {
   int i,j;
   for(i=1;i<=n;i++){
      for(j=1;j<=n;j++){
         if(i+j<=n)
         cout<<' ';
         if(i+j>n)
         cout<<'#';
       }
      cout<<'\n';
   }
}
```

14.https://www.hackerrank.com/challenges/birthday-cake-candles
Status:All test cases accepted
Code:

```
int birthdayCakeCandles(vector<int> candles) {

   int max = candles[0];
   int count =0;
   for(int i=0;i<candles.size();i++){
      if(candles[i]>max)
      max=candles[i];
   }

   for(int i=0;i<candles.size();i++){
      if(max==candles[i])
      count++;

   }
  return count;
}
```

15.https://www.hackerrank.com/challenges/the-hurdle-race
Status:All test cases accepted
Code:

```
#include <bits/stdc++.h>
using namespace std;
vector<string> split_string(string);

// Complete the hurdleRace function below.
int hurdleRace(int k, vector<int> height) {

   int max = height[0];
   int count=0;
   for(int i=0;i<height.size();i++){
      if(height[i]>max)
      max=height[i];
   }
   int n = abs(max-k);
   int p = k+n;

   for(int i=0;i<height.size();i++){
      if(k>=height[i])
```

```
        count=0;
        else if(k<height[i]){
        count =n;
        break;
        }
    }

    return count;
}
```