Name: Gaurav Yadav

Roll Number: 21075034

Dept: CSE(B.tech)

E-mail: gaurav.yadav.cse21@itbhu.ac.in

Lab – 6

Q1) Give an O(nlogk) time algorithm to merge k sorted lists into one sorted list where n is the total number of elements in all the input lists. (use min-heap)

```cpp
#include<bits/stdc++.h>
#define all(a) a.begin(),a.end()
#define rev(a) a.rbegin(),a.rend()
#define endl '\n'

using namespace std;
typedef long long ll;
const ll N = 1000000007;
const ll INF = 1000000000000000000;

int main(){
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    int k; cin >> k;
    vector<vector<int>> a(k + 1);
    int n = 0;
    for(int i = 1; i <= k; i++){
        int s; cin >> s;
        n += s;
        for(int j = 0; j < s; j++){
            int x; cin >> x;
            a[i].push_back(x);
        }
    }
    vector<int> res(n, 0);
    vector<int> ptr(k + 1, 0);
    vector<pair<int,int>> heap(k + 1);
    for(int i = 1; i <= k; i++){
        heap[i] = {a[i][ptr[i]], i};
    }
    int curr = 0;

    function<void(int)> heapify = [&](int u){
        int left = 2*u, right = 2*u + 1, mn = u;
        if(heap[left].first < heap[mn].first && left <= k){
            mn = left;
        }
        if(heap[right].first < heap[mn].first && right <= k){
            mn = right;
        }
        if(mn != u){
```

```
42              swap(heap[u], heap[mn]);
43              heapify(mn);
44          }
45      };
46
47      auto insert = [&](int val, int ind){
48          heap[k] = {val, ind};
49          int node = k;
50          while(node != 0){
51              if(heap[node].first < heap[node/2].first){
52                  swap(heap[node], heap[node/2]);
53                  node /= 2;
54              } else break;
55          }
56      };
57
58      for(int i = k/2; i >= 1; i--){
59          heapify(i);
60      }
61
62      while(k >= 0){
63          int num = heap[1].first;
64          int ind = heap[1].second;
65          res[curr++] = num;
66          swap(heap[1], heap[k]);
67          ptr[ind]++;
68          k--;
69          heapify(1);
70          if(ptr[ind] < a[ind].size()){
71              k++;
72              insert(a[ind][ptr[ind]], ind);
73          }
74      }
75      for(int &x: res) cout << x << " "; cout << endl;
76      return 0;
77  }
```

OUTPUT:

```
PS C:\Users\Gaurav\Programming\practice\cp> cd "c:\Users\Gaurav\Programming\practi
3
3
1 2 3
3
6 7 8
5
4 5 9 10 11
1 2 3 4 5 6 7 8 9 10 11
PS C:\Users\Gaurav\Programming\practice\cp> _
```

21075034 Gaurav Yadav

Q2) Write a program to find the minimum number of elements that must be swapped during the deletion of the maximum element from a max-heap of size N with no duplicate keys.

Time Complexity: O(n): for building the max-heap

O(logn): for calculating minimum number of swaps after deleting the maximum element.

```cpp
#include <bits/stdc++.h>
using namespace std;
vector<int> a, heap;
int heapify(int index, int &size)
{
    int ans = 0;
    int i = index;
    int left = 2 * i + 1;
    int right = 2 * i + 2;
    int largest = index;
    if (left < size && heap[left] > heap[i])
    {
        largest = left;
    }
    if (right < size && heap[right] > heap[largest])
    {
        largest = right;
    }
    if (i != largest)
    {
        swap(heap[i], heap[largest]);
        ans++;
        ans += heapify(largest, size);
    }
    return ans;
}
int Delete(int index, int &size)
{
    int ans = 1;
    swap(heap[index], heap[size - 1]);
    size--;
    ans += heapify(index, size);
    return ans;
}
int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    int n;
    cin >> n;
    a.resize(n);
```

21075034 Gaurav Yadav

```
42      heap.resize(n);
43      for (auto &i : a)
44          cin >> i;
45      for (int i = 0; i < n; i++)
46      {
47          heap[i] = a[i];
48      }
49      for (int i = (n - 1) / 2; i >= 0; i--)
50      {
51          heapify(i, n);
52      }
53      int heapsize = n;
54      int swaps = Delete(0, heapsize);
55      cout << "The minimum number of swaps are: " << swaps << '\n';
56      // swaps range from 1 to log2(n)+1
57      return 0;
58  }
```

Output:

```
PS C:\Users\Gaurav\Programming\practice> cd "c:\Users\Gaurav\Programming
9
9 2 1 4 5 6 11 8 3
The minimum number of swaps are: 3
PS C:\Users\Gaurav\Programming\practice\cp>
```