ASK/VIEW DOUBT | SOLUTION | HINT | Your Code | Main Code | Code Pair | C++ (g++ 5)

Problem | Result

Merge K sorted arrays

Send Feedback

Given k different arrays, which are sorted individually (in ascending order). You need to merge all the given arrays such that output array should be sorted (in ascending order).

Hint : Use Heaps.
Input Format :

Line 1 : Integer K
Line 2 : Arrays 1 size (n1)
Line 3 : Array 1 elements (separated by space)
Line 4 : Array 2 size (n2)
and so on for next arrays

Sample Input 1 :

```
4
3
1 5 9
2
```

PREVIOUS | NEXT

```cpp
#include<queue>
#include<vector>

using namespace std;


class triplet{
    public:
    int element;
    int ai;
    int ei;
};


class comp{
    public:
    bool operator()(triplet const * first, triplet const * second){
        return first->element > second->element;
    }
};

vector<int> mergeKSortedArrays(vector<vector<int>*> input){
    /* Don't write main().
     * Don't read input, it is passed as function argument.
     * Return the output.
     * Taking input and printing output is handled automatically.
     */
  /*  pair<pair<int,int>,int> p;
    pair<int,int> small;
    small.first=2;
    small.second=4;
    p.first=small;
    p.second=5;*/
    vector<int> output;

        triplet * temp = new triplet;
        temp->element = input.at(i)->at(0);
        temp->ai = i;
        temp->ei = 0;

        pq.push(temp);

    }

    while(!pq.empty()){

        triplet * ans = pq.top();
        pq.pop();
        output.push_back(ans->element);

        if(ans->ei+1 < (input.at(ans->ai)->size())){
            triplet * temp = new triplet;
            temp->element = input.at(ans->ai)->at(ans->ei+1);
            temp->ai = ans->ai;
            temp->ei = ans->ei+1;

            pq.push(temp);

        }

    }

    return output;

}
```

CUSTOM INPUT | SUBMIT SOLUTION