C++方向编程题答案

第一周

day1

> **100449-组队竞赛**
>
> 链接：https://www.nowcoder.com/questionTerminal/6736cc3ffd1444a4a0057dee89be789b?orderByHotValue=1&page=1&onlyReference=false
>
> **【题目解析】：**
>
> 队伍的水平值等于该队伍队员中第二高水平值，为了所有队伍的水平值总和最大的解法，也就是说每个队伍的**第二个值是尽可能大的值**。所以实际值把最大值放到最右边，最小是放到最左边。
>
> **【解题思路】：**
>
> - 本题的主要思路是贪心算法，贪心算法其实很简单，就是每次选值时都选当前能看到的局部最解忧，所以这里的贪心就是保证每组的第二个值取到能选择的最大值就可以，我们每次尽量取最大，但是最大的数不可能是中位数，所以退而求其次，取 每组中第二大的
> - 排序，然后取下标为3n - 2，3n - 4 ，3n - 4... n+2，n位置的元素累加即可，相当下标为[0,n-1]的n个数做每组的最左边的数，剩下的2个数据两个为一组，大的值做最右边的数，次大的做中间值，这里就是把这个次大的值加起来。
> - 例如 现在排序后 有 1 2 5 5 8 9 ，那么取 8 和 5相加等于 13

```
#include<iostream>
#include<algorithm>
#include<vector>
using namespace std;
int main()
{
    // IO型OJ可能会有多组测试用例，所以这里要持续接收输入多组测试用例。
    int n;
    while (cin >> n)
    {
        long long sum = 0;
        vector<int> a;
        a.resize(3*n);
        for (int i = 0; i < (3 * n); i++)
        {
            cin >> a[i];
        }

        /*
        排序，然后取下标为3n - 2，3n - 4 ，3n - 4... n+2，n位置的元素累加即可，
        相当下标为[0,n-1]的n个数做每组的最左边的数，剩下的2个数据两个为一组，
        大的值做最右边的数，次大的做中间值，这里就是把这个次大的值加起来
        */
        std::sort(a.begin(), a.end());
        for (int i = n; i <= 3 * n - 2; i += 2)
```

```
        {
            sum += a[i];
        }
        cout << sum << endl;
    }
}
```

69390-删除公共字符

链接：https://www.nowcoder.com/practice/f0db4c36573d459cae44ac90b90c6212?tpId=85&&tqId=29868&rp=1&ru=/activity/oj&qru=/ta/2017test/question-ranking

【题目解析】：

本题描述很简单，题目描述很清楚，读题即可

【解题思路】：

本题如果使用传统的暴力查找方式，如判断第一个串的字符是否在第二个串中，在再挪动字符删除这个字符的方式，效率为O(N^2)，效率太低，很难让人满意。

1. 将第二个字符串的字符都映射到一个hashtable数组中，用来判断一个字符在这个字符串。
2. 判断一个字符在第二个字符串，不要使用删除，这样效率太低，因为每次删除都伴随数据挪动。这里可以考虑使用将不在字符添加到一个新字符串，最后返回新新字符串。

```
#include<iostream>
#include<string>
using namespace std;

int main()
{
    // 注意这里不能使用cin接收，因为cin遇到空格就结束了。
    // oj中IO输入字符串最好使用getline。
    string str1,str2;
    //cin>>str1;
    //cin>>str2;
    getline(cin, str1);
    getline(cin, str2);

    // 使用哈希映射思想先str2统计字符出现的次数
    int hashtable[256] = {0};
    for(size_t i = 0; i < str2.size(); ++i)
    {
        hashtable[str2[i]]++;
    }

    // 遍历str1，str1[i]映射hashtable对应位置为0，则表示这个字符在
    // str2中没有出现过，则将他+=到ret。注意这里最好不要str1.erases(i)
    // 因为边遍历，边erase，容易出错。
    string ret;
    for(size_t i = 0; i < str1.size(); ++i)
    {

        if(hashtable[str1[i]] == 0)
```

```
            ret += str1[i];
        }

        cout<<ret<<endl;
        return 0;
    }
```