**每日一题总结**

Day1

100449-组队竞赛

/\*本题的主要思路是贪心算法，保证每组的第二个值取到最大就  
\* 排序，然后取 第 3n - 1 3n - 3 3n - 5...个元素累加即可  
\* 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  
\* 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  
\*/  
#include<iostream>  
#include<algorithm>  
#include<vector>  
using namespace std;  
int main()

{  
 int n;  
 while (cin >> n)

{  
 long long sum = 0;  
 vector<int> a(3 \* n);  
 for (int i = 0; i < (3 \* n); i++)

{  
 cin >> a[i];  
 }   
 sort(a.begin(), a.end());  
 for (int i = n; i <= 3 \* n - 2; i += 2){  
 sum += a[i];  
 }   
 cout << sum << endl;  
 }  
}

删除公共字符

#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;  
}

Day2

排序子序列

#include<iostream>  
#include<vector>  
using namespace std;  
int main()  
{  
int n;  
cin>>n;  
vector<int> v;  
v.resize(n);  
for(int i = 0; i < n; ++i)  
cin>>v[i];  
//找出波峰波谷  
int ret = 1;  
for(int i=1; i<n-1; i++)  
{  
if((v[i-1] < v[i] && v[i] > v[i+1]) || (v[i-1] > v[i] && v[i] < v[i+1]))  
{  
ret++;  
// 如果发现波峰或者波谷，这里i++已经比过了，则需要跳过，但是要注意越界。  
if(i != n-3)  
i++;  
}  
}

cout<<ret<<endl;  
}

倒置字符串  
https://www.nowcoder.com/practice/ee5de2e7c45a46a090c1ced2fdc62355?tpId=85&&tqId=29867&rp=1&ru=/activity/oj&qru=/ta/2017test/question-ranking  
#include<iostream>  
#include<vector>  
using namespace std;  
int main()  
{  
int n;  
cin>>n;  
vector<int> v;  
v.resize(n);  
for(int i = 0; i < n; ++i)  
cin>>v[i];  
//找出波峰波谷  
int ret = 1;  
for(int i=1; i<n-1; i++)  
{  
if((v[i-1] < v[i] && v[i] > v[i+1])  
|| (v[i-1] > v[i] && v[i] < v[i+1]))  
{  
ret++;  
// 如果发现波峰或者波谷，这里i++已经比过了，则需要跳过，但是要注意越界。  
if(i != n-3)  
i++;  
}  
}

cout<<ret<<endl;  
}

#include <iostream>  
#include <string>  
#include <algorithm>  
using namespace std;  
or  
int main()  
{  
string s;  
getline(cin, s));  
// 翻转整个句子  
reverse(s.begin(), s.end());  
// 翻转单词  
auto start = s.begin();  
while (start != s.end())  
{  
auto end = start;  
while (end != s.end() && \*end != ' ')  
end++;  
reverse(start, end);  
if (end != s.end())  
start = end + 1;  
else  
start = end;  
}

cout << s << endl;  
return 0;  
} #  
include <iostream>  
#include <string>  
using namespace std;  
// cin读取string时自动会被空格分隔开，用另一个字符串存储进行逆序输出  
int main()  
{  
string s1, s2;  
cin >> s2;  
while (cin >> s1)  
s2 = s1 + " " + s2;  
cout << s2 << endl;  
return 0;  
}

**day3**  
题目ID：69385--字符串中找出连续最长的数字串  
链接：https://www.nowcoder.com/practice/bd891093881d4ddf9e56e7cc8416562d?tpId=85&&tqId=29864&rp=1&ru=/activity/oj&qru=/ta/2017test/question-ranking  
#include<iostream>  
#include<string>  
using namespace std;  
int main()  
{  
string str,res,cur;  
cin>>str;  
for(int i=0;i<=str.length();i++)  
{  
// 数字+=到cur  
if(str[i]>='0' && str[i]<='9')  
{  
cur+=str[i];  
}

else  
{  
// 找出更长的字符串，则更新字符串  
if(res.length() < cur.length())  
res=cur;  
else  
cur.clear();  
}  
}

cout<<res;  
return 0;  
}

69387-n个数里出现次数大于等于n/2的数  
https://www.nowcoder.com/practice/eac8c671a0c345b38aa0c07aba40097b?tpId=85&&tqId=29866&rp=1&ru=/activity/oj&qru=/ta/2017test/question-ranking  
  
//O(n)思想：因为要找过半的数，用一个变量count记录读取每个变量变化的次数，  
//一个变量temp记录可能过半的数，先让count=1，然后让temp=vec[0]，  
//然后往后遍历一遍，碰到和temp相同的数就给count++，否则就count--  
//如果，count变成0，就让temp=vec[i](vec数组遍历过程当前值),并让count=1  
//如此遍历一遍，因为有一个数过半，所以temp最后肯定存储的是过半的数  
#include <vector>  
#include <iostream>  
or  
using namespace std;  
int main()  
{  
int n;  
vector <int> vec;  
while(cin >> n)  
vec.push\_back(n);  
int count = 1;  
int temp = vec[0];  
for(int i = 1; i < vec.size(); ++i){  
if(vec[i] == temp)  
count++;  
else  
count--;  
if(count == 0)  
{  
temp = vec[i];  
count++;  
}  
}

cout << temp << endl;  
return 0;  
}

#include<iostream>  
using namespace std;  
int main(){  
int count[100]={0};//数组置0，然后输入n，则对应位置加1，最后遍历数组  
int x = 0, n = 0; //找到大于等于一半次数的那个  
while(cin>>x){  
count[x]++;  
n++;  
}

for(int i=0; i<n; i++){  
if(count[i] >= n/2)  
cout<<i<<endl;  
}

return 0;  
}

**day4**  
题目ID：46579 --计算糖果

https://www.nowcoder.com/practice/02d8d42b197646a5bbd0a98785bb3a34?tpId=85&&tqId=29857&rp=1&ru=/activity/oj&qru=/ta/2017test/question-ranking  
#include<iostream>  
using namespace std;  
// A - B = a  
// B - C = b  
// A + B = c  
// B + C = d  
// 这道题目的实质是：判断三元一次方程组是否有解及求解。  
// 这里是小学生都会的问题了^^  
int main(){  
int a,b,c,d;  
cin>>a>>b>>c>>d;  
int A=(a+c)/2;  
int C=(d-b)/2;  
int B1=(c-a)/2;  
int B2=(b+d)/2;  
if(B1!=B2){  
cout<<"No";  
}   
else{  
cout<<A<<" "<<B1<<" "<<C;  
}   
return 0;  
}

58541-进制转换  
https://www.nowcoder.com/practice/ac61207721a34b74b06597fe6eb67c52?tpId=85&&tqId=29862&rp=1&ru=/activity/oj&qru=/ta/2017test/question-ranking  
  
#include <iostream>  
#include<string>  
#include<algorithm>  
using namespace std;  
int main()  
{  
string s, table="0123456789ABCDEF";  
int m,n;  
cin>>m>>n;  
bool flag = false;  
// 如果是负数，则转成正数，并标记一下  
if(m < 0)  
{  
m = 0 - m;  
flag = true;  
} /  
/ 按进制换算成对应的字符添加到s  
while(m)  
{  
s += table[m%n];  
m /= n;  
}   
if(flag)  
s += '-';  
reverse(s.begin(), s.end());  
cout<<s<<endl;  
return 0;  
}