

CF Edu Round 93

A

题意：

给你一个序列，问这些序列中取三个数字无法构成三角形的元素，随意一组，如果都可以组成，那么输出 -1。

思路：

三角形两条边的和必大于第三条边，那么我们直接将两个最小的加起来直接和最大进行相比，如果这个可以构成三角形的话，那么这个里面都可以构成三角形。

```
#include<iostream>
#include<cstdio>
#include<cmath>
#include<cstring>
#include<algorithm>
#include<vector>
#include<queue>
using namespace std;
int pic[100050];
int main()
{
    int t = 0;
    cin >> t;
    while (t--)
    {
        int n;
        cin >> n;
        for (int i = 1; i <= n; i++)
            cin >> pic[i];
        if (pic[1] + pic[2] <= pic[n])
            printf("1 2 %d\n", n);
        else
            puts("-1");
    }
    return 0;
}
```

B

题意：

有两个人，每一次都可以取序列中连续的部分，然后取完的两边会相邻，问先取的那个人最多可以取多少个1，两个人都取最优。

思路：

直接将连续的1个数全部提取出来，先取的是拿最大的，后取的第二大直到取完。

```
#include<iostream>
#include<cstdio>
#include<cmath>
#include<cstring>
#include<algorithm>
#include<vector>
#include<queue>
using namespace std;
char s[1024];
int num[1024], tot;
int main()
{
    int t;
    cin >> t;
    while(t--)
    {
        cin >> s;
        tot = 0;
        int tem = 0;
        for (int i = 0; s[i];i++)
        {
            if (s[i] == '1')
                tem++;
            else
            {
                if (tem != 0)
                {
                    num[tot++] = tem;
                    tem = 0;
                }
            }
        }
        if(tem != 0)
            num[tot++] = tem;
        sort(num, num + tot);
        int ans = 0;
        for (int i = tot - 1; i >= 0;i-=2)
            ans += num[i];
    }
}
```

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

C

题意：

在一个序列中，我们需要求出有多少个区间满足 $\sum_{i=l}^r a_i = r-l+1$

```
#include <algorithm>
#include <cmath>
#include <cstdio>
#include <cstring>
#include <iostream>
#include <queue>
#include <vector>
using namespace std;
typedef long long LL;
const int MAX = 1e5 + 20;
int a[MAX];
int sum[MAX];
int cnt[2000090];
char s[MAX];
LL ans = 0;
int main()
{
    int t;
    cin >> t;
    while (t--)
    {
        int n;
        ans = 0;
        cin >> n;
        cin >> s;
        for (int i = 0; i < n; i++)
        {
            a[i] = s[i] - '0';
            sum[i] = a[i];
        }
        for (int j = 1; j < n; j++)
            sum[j] += sum[j - 1];
        for (int i = 0; i < n; i++)
            sum[i] -= i + 1;
        for (int i = 0; i <= n * 10 * 2 + 2; i++)
            cnt[i] = 0;
    }
}
```

```
    cnt[0 + n * 10] = 1;
    for (int i = 0; i < n; i++)
    {
        if (cnt[sum[i] + n * 10])
            ans += cnt[sum[i] + n * 10];
        cnt[sum[i] + n * 10]++;
    }
    cout << ans << endl;
}
}
```

D

题意：

给你三个序列，你每次可以选择三个序列中的两个元素，但不能是相同序列的，然后进行相乘，问最后的相乘的和最大是多少。

题解：

先大到小排序，然后暴力三维dp即可 $dp[i][j][k]$ 表示r,g,b分别选到第i,j,k个时的最大值。

```
#include<iostream>
#include<cstdio>
#include<cmath>
#include<cstring>
#include<algorithm>
#include<vector>
#include<queue>
using namespace std;
typedef long long ll;
int R[256], G[256], B[256];
int r, g, b;
ll dp[256][256][256];
ll ans;
void mfs(int x,int y,int z)
{
    if ((x > r and y >g) or( z >b and y >g) or (x>r and z>b))
    {
        dp[x][y][z] = 0;
        return;
    }

    if (dp[x][y][z] != 0)
        return;
```

```

if (x<=r and y<=g)
{
    mfs(x + 1, y + 1, z);
    dp[x][y][z] = max(dp[x][y][z], dp[x + 1][y + 1][z] + R[x] * G[y]);
}
if (x<=r and z<=b)
{
    mfs(x + 1, y, z+1);
    dp[x][y][z] = max(dp[x][y][z], dp[x + 1][y][z + 1] + R[x] * B[z]);
}
if (y<=g and z<=b)
{
    mfs(x, y + 1, z+1);
    dp[x][y][z] = max(dp[x][y][z], dp[x][y + 1][z + 1] + B[z] * G[y]);
}
//dp[x][y][z] = max(dp[x + 1][y + 1][z] + R[x] * G[y], max(dp[x +
1][y][z + 1] + R[x] * B[z], dp[x][y + 1][z + 1] + B[z] * G[y]));
ans = max(ans, dp[x][y][z]);
}
bool cam(int x,int y)
{
    return x > y;
}
int main()
{
    scanf("%d%d%d", &r, &g, &b);
    //memset(dp, -1, sizeof(dp));
    for (int i = 1; i <= r;i++)
    {
        int tem;
        scanf("%d", &tem);
        R[i] = tem;
    }
    for (int i = 1; i <= g;i++)
    {
        int tem;
        scanf("%d", &tem);
        G[i] = tem;
    }
    for (int i = 1; i <= b;i++)
    {
        int tem;
        scanf("%d", &tem);
        B[i] = tem;
    }
    sort(R + 1, R + r + 1, cam);
    sort(G + 1, G + g + 1, cam);
    sort(B + 1, B + b + 1, cam);
    mfs(1, 1, 1);
    cout << ans << endl;
}

```

```
return 0;  
}
```

E

题意：

我们总共有两种法术，闪电法术、火焰法术，闪电法术之后能够将下一个法术的伤害翻倍，现在有两种操作学习一种法术/忘记一种法术，问你每每次操作后最大的伤害是多少。

题解：

贪心set乱搞。

From:
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:too_low:cfedu93cy&rev=1598003448

Last update: **2020/08/21 17:50**