

2925---教主的花园 (garden)

【算法步骤】:

1. 先把 $a[i]$ 进行快排
2. 对于每个询问分别处理;
 - ① 特殊情况: 在同一边的情况
 - ② 一般情况: 先二分查找小于等于 $x1$ 的点 $a[k]$;
 - ③ 再考虑 $a[k]$ 和 $a[k+1]$ 两个入口对结果的影响;

2926---教主泡嫦娥

(goddess)

【题目简述】环形山上有 N 个点，相邻点高度不同，你可选择顺时针或者逆时针走，任何时刻处于“上升”或“下降”中的一种，起始点可以任意选择。若从第 i 个点，走到第 j 个点（ i 和 j 相邻）

j 点高于 i 点：若处于上升状态，需要耗费两段高度差的绝对值的体力；否则耗费高度差平方的体力。

j 点低于 i 点：若处于下降状态，需要耗费两段高度差的绝对值的体力；否则耗费高度差平方的体力。

在每一个点时，可以选择切换自己的状态（上升→下降，下降→上升），每次切换需要耗费 M 点的体力。在起点时，可以选择任意状态并且不耗费体力

问走一圈所消耗的最小体力值。

【题目考点】递推型 DP

首先，相邻的两个点之间只有 4 种情况（00,01,10,11）

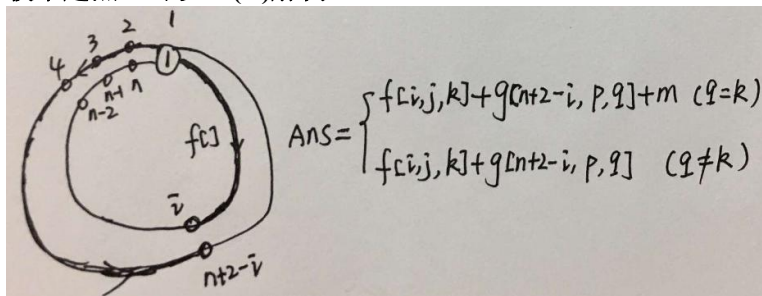
简化问题：固定起点 1 顺时针走到 1，如何计算？----直接递推计算即可

先初始化， $f[i,j,k]$ 表示顺时针第一个点状态为 k , i 点状态为 j , 所耗费的最小体力

本题起点任意，枚举起点，时间复杂度 $O(n^2)$ ，超时

$g[i,j,k]$ 表示逆时针第一个点状态为 k , i 点状态为 j , 所耗费的最小体力

枚举起点，可以 $O(N)$ 解决

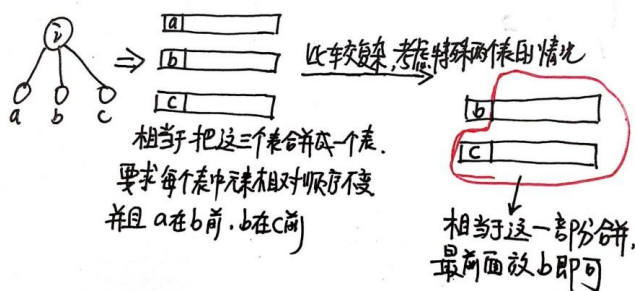


}

2927---保镖排队

(lineup.pas/c/cpp)

【方法 1】：树型 DP+组合数学



状态: $f[x]$ 表示以 x 为根结点的子树有多少个排列满足要求。

转移: $f[i]=1$;

$s=0$;

For i 的所有子结点 k (输入顺序逆序, 即从大到小)

{ DP(k); // 递归求解子结点

$s=s+\text{son}[k]$;

$f[i]=f[i]*C(s, \text{son}[k])*f[k]$;

$s=s+1$; // k 结点

}

【方法 2】

先转为二叉树, DFS 求出每个子树结点个数, 最后树形 DP

因为当树转为二叉树后会发现, 只需要满足父亲结点在儿子结点前面即可, 于是设 $f[i]$ 表示以 i 为根的子树合法的方案总数, 则:

$F[i]=f[i.L]*f[i.R]*C(\text{son}[i.L]+\text{son}[i.R], \text{son}[i.L])$

$\text{Ans}=f[1]$ 。

2928---教主的别墅 (villa)

【方法 1】资源分配类 DP，期望得分 45 分。

```
#include<iostream>
#include<cmath>
using namespace std;
int n,m,a[1005]={0},s[1005]={0};
int f[1005][1005]={0};
int b[1005]={0};
void init()
{ int i,j;char c;
  cin>>n>>m;s[0]=0;
  for(i=1;i<=n;i++)
  { cin>>c;
    a[i]=c-'0';
    s[i]=s[i-1]+a[i];
  }
}
void solve()
{ int i,j,k,x,lst,bj=0;
  for(i=1;i<=m;i++)
    for(j=1;j<=n;j++)f[i][j]=0x7fffffff/2;
  for(i=1;i<=n;i++)f[1][i]=abs((i-s[i])-s[i]);
  for(i=2;i<=m;i++)
    for(j=i;j<=n;j++)
      for(k=i-1;k<j;k++)
        { x=max(f[i-1][k],abs((j-k)-(s[j]-s[k])-(s[j]-s[k])));
          if(f[i][j]>x)f[i][j]=x;
        }
  lst=n;
  for(i=m;i>=2;i--)
  { bj=0;
    for(j=i-1;j<lst;j++)
      if(max(f[i-1][j],abs((lst-j)-(s[lst]-s[j])*2))<=f[m][n]&&!bj)bj=j;
    b[i]=lst-bj;lst=bj;
  }
  cout<<lst<<" ";
  for(i=2;i<=m;i++)cout<<b[i]<<" ";cout<<endl;
  lst=n;
  for(i=m;i>=2;i--)
  { bj=0;
    for(j=lst-1;j>=i-1;j--)
      if(max(f[i-1][j],abs((lst-j)-(s[lst]-s[j])*2))<=f[m][n]&&!bj)bj=j;
    b[i]=lst-bj;lst=bj;
  }
  cout<<lst<<" ";
  for(i=2;i<=m;i++)cout<<b[i]<<" ";
}
int main()
{ init();
  solve();
  return 0;
}
```

【方法 2】贪心。

先求出男女生的差值尽量平均分配到 m 个房间，用 $O(n)$ 的时间扫一遍即得答案。注意：差值

为 0（即男女生人数一样的时候）要特殊处理。

■ $a[i]$ 1 1 0 0 1 1 0 0

■ $sum[i]$ 1 2 1 0 1 2 1 0---男-女的值

■ 先求出男女的差值，尽量平均分配到 m 个房间

■ 若 m 大于 $sum[]$ 中 0 的个数并且 $sum[n]=0$ ，则 $Ans=0$ ；

■ 否则 $Ans=|sum[n]-1|/m+1$ ；

■ 最小方案：从左到右扫描，使得剩下尽量多

■ 若 $(abs(sum[n]-sum[i]) \leq ans * (m - tot - 1))$ 成立，则可行

最大值，只需要将序列倒序，然后求最小值即可

$a[i]$ 0 0 1 1 0 0 1 1