算法设计与分析　　2020K8009907032　唐嘉良

1. (1) 问题描述：给定一个2维数组，每个元素是一个存储高和宽的邮票数据。一枚邮票能嵌入另一枚邮票当且仅当这枚邮票的高和宽分别比另一枚邮票的高和宽要小。在不能旋转的前提下，计算最多能套娃多少张邮票。

(2) 分析：要使得套娃的邮票最多，则要让外层邮票尽可能大。我们每次从这些邮票中选出最大的一张即可。我们给所有邮票按宽+高的大小顺序排序，从大到小选出邮票，每次要判断是否可以套进去。

```
Russian_Roll (envelope[]){
    ~~for i from 0 to len(envelope-1)~~
    sort(envelope, envelope+len(envelope), cmp); //排序
    count=0; now_h=0; now_w=0;
    for i from 0 to len(envelope)-1
        if i==0
            count++; now_h = envelope[i].height; now_w = envelope[i].width;
        else ~~if i==len~~
            if envelope[i].height < now_h && envelope[i].width < now_w
              {count++;
               now_h = envelope[i].height;
               now_w = envelope[i].width;}
    return count;
}
cmp (*x, *y){
    ~~return (x+y > y+x);~~
    return (x.height+x.width > y.height+y.width);
}
```

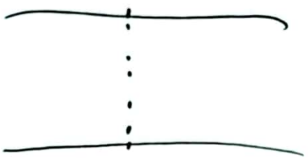
正确性分析：首先考虑到一个事实：如果某个邮票宽、高之和大于当前套娃的最小邮票，则不可以套进去（因为宽和高必然有一个比最小邮票对应的宽和高要大，否则不满足序关系），而且第一张邮票必选，否则数量只会更少，后续按宽+高的序关系找下去即可，正确性得证。

(3) 设数组长、宽为 m, n，排序要花 $O(mn\log(mn))$，扫一遍数组找满足的邮票要花 $O(mn)$，总计时间复杂度 $O(mn\log(mn))$

只需要所给的二维数组和几个变量存当前最小长宽，一个变量计数。

总计空间复杂度 $O(mn)$.

2.



(1) 利用数组dp[]来存放跳到第i个石头最后一步的跳跃距离

则 dp[0]=0, dp[1]=(stone[1]-stone[0]<=1)?1:stone[1]-stone[0],

dp[i]=(stone[i]-stone[i-1]>=dp[i-1]+1 &&
stone[i]-stone[i-1]>=dp[i-1]-1)? stone[i]-stone[i-1]:-1;

即: dp[0]跳到第0个石头,无需步长

dp[1]跳到第1个石头,由于第1步只能跳一个距离,判断第0个和第1个石头距离是否小于等于1,是则更新为两石头距离,否则为-1(跳不到).

为两石头距离是否落在[k-1,k+1]内即可.最后看dp[n-1]正负号即可.

(2) StoneQ_jump(stone[], n){

for(i from 0 to n-1

if i==0

dp[i]=0

if i==1

dp[i]=(stone[1]-stone[0]<=1)?stone[1]-stone[0]:-1;

if i>=2

dp[i]=(stone[i]-stone[i-1]<=dp[i-1]+1 &&
stone[i]-stone[i-1]>=dp[i-1]-1)?stone[i]-stone[i-1]:-1;

return (dp[n-1]>=0)?'TRUE':'FALSE';

}

(3) 只需扫一遍数组来更新dp,时间复杂度O(n)

只需用线性O(n)长的数组dp记录,空间复杂度O(n).

注:如果允许跨石头跳跃,则dp应修改为对此前所有石头到本石头距离与此前石头处dp值两大小关系判断,加一层循环,时间复杂度为$O(n^2)$,空间复杂度不变.

2

3. 给一个带权边的树有从1到n标号的n个结点，任两结点间只有唯一路径，且该路径上所有边的权重的异或值叫特征分，设计一个算法找到最大的特征分.

(2) 分治算法，找此时根结点为起点或终点时的最大特征分，分为若干子树，找这些子树中根结点为起点或终点时最大特征分，看加入根结点所在边后是否更优，同时需判断只有根边的情况. 对所有结点重复上面操作，即可找到所有path.

```
MaxScore(*tree){
    input(*tree); //处理输入，注意边数=点数-1，用一个循环来读边信息即可，组织成一棵树
max = Calmax(*tree);
    return max;
}
Calmax(*tree){
    if(*tree==NULL)
        return 0;
    if(*tree->left==NULL)
        max1=
        Calmax(tree->right);
        max=(max1⊕right_edge>max)?max1⊕right_edge:max;
    score[n];
Max(*n){
    input();处理输入
    for each node t:
      score[n]=findMax(*t)
    max=max(score)
    return max;
}
findMax(*t){
    s[n]*n];
    for each edge connected to *t:
        s[n]=findMax(edge上另一顶点)⊕edge;
    if 叶子结点
        return 0;
    for each edge connected to *t:
        s[i++]=edge;
    res=max(s);
    return res;
}
```

(3) 对每个结点做一次，每次$O(n\log n)$

总计复杂度$O(n^2\log n)$.

空间复杂度$O(n)$.

4. 小设运营成本为使用运输车的成本，设I、II、III长度为l_1, l_2, l_3（$l_1<l_2$且$l_1<l_3$），设1-1车有n辆，1-2车有m辆（$m \le \beta \cdot n$）。设x_{ijhkw}表示第i类车第j辆放在第h类车第k辆的w层（w=0下层，w=1上层）。设每层长度为L。两类运输每输成本为c_1, c_2，待运车数目w_1, w_2, w_3.

目标函数：$\min \quad c_1 f_1 + c_2 f_2$，其中$f_1, f_2$是用到的1-1、1-2车数目，

$$f_1 = \sum_k g_{1,k} \qquad f_2 = \sum_k g_{2,k}$$

$$g_{1,k} = \begin{cases} 1, & \text{if } \sum_j \sum_w x_{ij1kw} > 0 \\ 0, & \text{else} \end{cases} \qquad g_{2,k} = \begin{cases} 1, & \text{if } \sum_j \sum_w x_{ij2kw} > 0 \\ 0, & \text{else} \end{cases}$$

(2) $f_1 \le n$

$f_2 \le m$

$$\sum_j (l_1 x_{1jhkw} + l_2 x_{2jhkw} + l_3 x_{3jhkw}) + (x_{1jhkw} + x_{2jhkw} + x_{3jhkw} - 1) \cdot \beta \le L \quad \text{for all } h, k, w \quad \text{——①}$$

分析：要保证车全运走，一层满才能装入2层，层满的判断如式①，此外还应保证每辆车只能放进运输车的唯一位置，且车只能放入1-1车的低层

2层必须两列车数相等

5.

(1) 设 X_{ij} 表示是否走了 V_i 到 V_j 的路，$X_{ij}=0$ 反之

目标函数：$\min \sum_{ij} X_{ij} D_{ij}$

假设从 V_1 出发.

(2) $\sum_i X_{ij} = 1$ for all j

$\sum_j X_{ij} = 1$ for all i

} 每点只走一次且不折返

$X_{ij} = 0/1$ for all i,j.

$X_{ij} + X_{ji} = 2/0$ for all ij. 线路选择对称

$X_{ii} = 0$ for all i.

~~$\sum X_{1j} = 1$ 出发~~

~~$\sum X_{i1} = 1$ 回来~~

6.

流 x/c

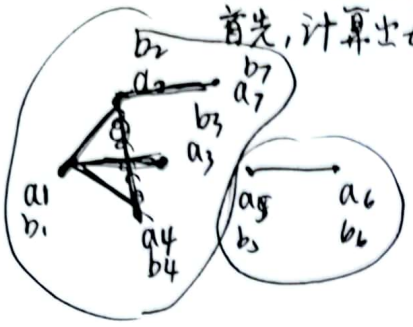
$a(x)$ 花费

算法描述：用网络流算法（FF）在每次选增广路时选择增广费用最小的一条路即可. 这样保证最终得到的是最大流，同时拥有最小费用.

或者先用网络流计算出最大流，再利用最大流，抽象出费用流网络，在费用网络中运用最小费用流算法.（对偶问题）转化为

7.

构建一个无向图



首先，计算出每个 $b_i - a_i$，设为 c_i，则 $\sum c_i = 0$，若不成立则无法满足（流守恒）

（若允许每个客户动无数步，则仅需判断不 $c_i = 0$ 即可，若成立，则一定满足）

对每个连通子图，~~从第一个开始~~

对每条边设一个流 x_{ij} 满足 $x_{ij} = -x_{ji}$

对每个顶点列出方程，得到方程组，解出答案。若无解则不满足

若解合乎题意则满足（点与边的数量保证了是可解的）

或无符合题意而解