

# 加分二叉树

## 一. 考察内容:

动态规划 树形DP

## 二. 题目分析:

[题目大意]

定义一颗二叉树的加分为其左右加分乘积+该点权值，给定一棵树的中序遍历和每个点的权值，要求构造一棵加分最大的树。

[写题思路]

考虑从子树开始构造，设状态 $f[i][j]$ :  $i-j$ 组成的子树的最大权值。

转移：对于一颗 $i-j$ 构成的子树，他可以有二个儿子，所以我们枚举一个点，这个点左边的属于左儿子，剩余的属于右儿子，那么该子树的加分就是左儿子的加分\*右儿子的加分+该点权值，最后只需求出 $f[1][n]$ 即可。

## 三. 代码实现:

```
#define _CRT_SECURE_NO_DEPRECATE
/*****
*创建时间: 2018 08 07
*文件类型: 源代码文件
*题目来源: 洛谷
*当前状态: 已通过
*备忘录: 动态规划 区间DP
*作者: HtBest
*****/
#include <stdio.h>
#include <iostream>
#include <string>
#include <stdlib.h>
#include <string.h>
#include <algorithm>
#include <queue>
using namespace std;
#define MAXN 35
int n,a[MAXN],f[MAXN][MAXN],root[MAXN][MAXN];
/* Variable explain:

*/
void read()
{
    // freopen(".in","r",stdin);
    // freopen(".out","w",stdout);
    scanf("%d",&n);
    for(int i=1;i<=n;++i)scanf("%d",&a[i]);
    return;
}
void print(int l,int r)
{
    int lson=root[l][root[l][r]-1],rson=root[root[l][r]+1][r];
    printf("%d ",root[l][r]);
    // printf("[%d,%d]:%d\n",l,r,root[l][r]);
    if(lson)print(l,root[l][r]-1);
    if(rson)print(root[l][r]+1,r);
}
void dp()
{

```

```
//状态: f[i][j]:由节点i~j构成的子树最大加分
int ans=0;
for(int i=1;i<=n;++i)//长度
{
    for(int j=1;j<=n-i+1;++j)//起点
    {
        for(int k=j;k<=j+i-1;++k)//root
        {
            if(i==1){f[j][j]=a[j];root[j][j]=j;continue;}
            if(f[j][j+i-1]<max(1,f[j][k-1])*max(1,f[k+1][j+i-1])+a[k])
            {
                f[j][j+i-1]=max(1,f[j][k-1])*max(1,f[k+1][j+i-1])+a[k];
                root[j][j+i-1]=k;
            }
        }
    }
}
printf("%d\n",f[1][n]);
print(1,n);
}
int main()
{
    read();
    dp();
    return 0;
}
```

[<题目跳转>](#) [<查看代码>](#)