p1270

【算法分析】

然后考虑一个简单的dp[i][j]表示到第i条走廊剩余j的时间最多能拿的画的数量。

那么枚举一下分给左儿子的时间k，那么分给右儿子的就是j-k了。

所以dp[i][j]=max(dp[i][j],dp[lch][k]+dp[rch][j-k])

当然，还要减去一个经过走廊的时间。注意，走廊是要走过去并且走回来的，所以时间在读入时\*2就行了。注意可以不把一个展厅的画取完

注意警察一到小偷就死，所以小偷必须在警察来的前1秒跑掉。

建树是后序遍历，DP也是后序遍历，二者的结构有着惊人的相似之处，所以，不用建树。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | #include <bits/stdc++.h>  using namespace std;  int tot, n=0, dp[666][666]; // dp[i,j]表示用j秒回到i最多拿几幅画  void dfs(int root)  {  int time, pic;  cin>>time>>pic;  time \*= 2; // 走廊要走两遍  if (!pic) // 如果不是叶节点  {  int lc=++n, rc=++n; // 后序遍历  dfs(lc);  dfs(rc);  for (int i=time; i <= tot; ++i)  for (int j=0; j <= i - time; ++j) // 从去左边j秒，去右边可用时间减j秒  dp[root][i] =max(dp[root][i], dp[lc][j] + dp[rc][i - time - j]);  }  else  for (int j=time; j <= tot; ++j)  dp[root][j] = min((j - time) / 5, pic);//统计j秒能取多少画  }  int main()  {  cin>>tot;  dfs(0);  cout<<dp[0][tot]<<endl;  return 0;  } |