

Algorithm Homework1

PB18111704 Zhu Enzuo

2020 年 11 月 24 日

0.1 Prob1

状态: $dp[N]$ 表示恰好切掉长度为 C 的钢条的情况下最大收益为 $dp[C]$
转移: $dp[i]_{i=1\dots N-1} = \max(dp[i], (dp[i - l_j] > 0)dp[i - l_j] + p_j - c)$
 $dp[N] = \max_{j=1\dots M}(dp[N - l_j] + p_j)$
结果: $ans = \max_{(i = 1\dots N)}(dp[i])$

0.2 Prob2

Algorithm 1 Shortest Path

procedure SHORTESTPATH($nodeS, nodeT$)

if S 到 T 的最短路已经计算得出 **then**

return ShortestPathmap[S][T]

else

for each S's edge's target **do**

$ShortestPathmap[S][T] = \max(ShortestPathmap[S][T], ShortestPathmap[S][S - > next] + ShortestPath(S - > next, T))$

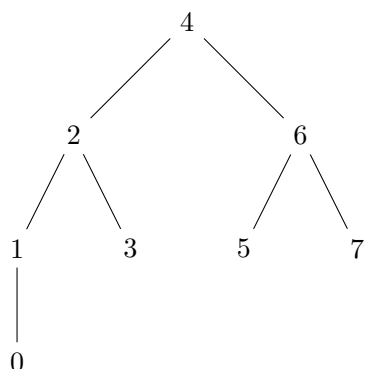
end for

end if

end procedure

0.3 Prob3

minimum: 1.22



0.4 Prob4

考虑最优子结构：

对于一个子树的根节点：由于子树的根节点这个员工的选择情况会影响到其父节点的选择状况。所以需要对根节点的选择情况计算保存两个不同的值。

Algorithm 2 BestChoise

procedure DFS(*noderoot*)

if root has no son **then**

$root \rightarrow mvwithroot = root \rightarrow value$

$root \rightarrow mvwithoutroot = 0$

return

end if

for each son of root **do**

 DFS(*son*)

end for

$root \rightarrow mvwithroot = \Sigma s \rightarrow mvwithoutroot + root \rightarrow value$

$root \rightarrow mvwithoutroot = \Sigma \max(s \rightarrow mvwithroot, s \rightarrow mvwithoutroot) + root \rightarrow value$

end procedure

时间复杂度：对于每一个节点只会自己计算的时候访问一次，算其父节点的时候被访问一次，所以每个节点都只会被访问 2 次，且每次访问都是 $O(1)$ 的。故总复杂度为 $O(n)$

0.5 Prob5

算法：先将所有实数从小到大排序。然后进行贪心：每一次取尚未被包含于区间中的值最小的点作为新区间的下界。然后将所有被这个区间包裹的点从原集合中删除。这样构造出来的一个单位区间集合就是最终的结果。

正确性：对于每一次选择的区间，由于最终结果一定会需要将当前最小的点包裹在区间中，而且如果移动区间使得有一些点不再属于这个区间，则之后的答案一定不优于当前结果。故这个算法是正确的。

0.6 Prob6

算法：从最大的硬币开始尝试：如果能找则直接用之，否则用小一点的开始尝试。

正确性：对于少找了 d 个 25 硬币，至少需要 $3d$ 个硬币才能更优。