计算机算法设计与分析

第 2 次作业

刘炼

202128013229021

Problem 1

Assignment Problem1

Solution

根据题目的描述，考虑两种不同的情况，也就是是否 all house are arranged in a circle; 其中假设第 i 家拥有的钱的数目为 $P(i)$，总的家庭的数目为 n。首先考虑不为 circle 的情况，则设从 i 到 j 家中，在安全情况下打劫的总数目为 $R(i\ j)$，由于相邻的家不能被打劫，所以，相应的最优子结构为

$$R(i,j) = \max\{R(i,j-1), R(i,j-2) + P(j)\} \tag{1}$$

当为 circle 的时候，则有一个特殊的考虑，即不能同时抢劫第 1 家和最后一家（第 n 家）。只有一个地方特殊，即应该有

$$R(1,n) = \max\{R(1,n-1), R(2,n)\} \tag{2}$$

也就是假设将第一家和最后一家中的一个去掉，让 circle 不成立，这样，变相变成了之前的问题的两倍，然后最终进行比较。

Pseudo-Code

由于实际上，根据上述分析，对于 circle 的情况，可以看成两个不是 circle 的情况然后进行比较，所以这里只给出不为 circle 情况下的伪代码

Pseudo-Code shown as follows.

Algorithm 1 Maximum Amount of Money

Input: P, n ▷ P is a non-negative integers array, n is the length of array.

```
function MAXMONEY(T, R, id)
    return max{R[id − 1], R[id − 2] + T[id]}
end function
Initialization: initialize an array R, R[0] = 0, R[1] = P[1]
for id ← 2 to n do
    R[id] ← MAXMONEY(T, R, id)
end for
```

Output: Maximum amonut of money $R[n]$

Proof of the correctness

使用结构归纳法证明如下：

- Initialization：对于长度为 1 的 hosue，直接抢劫后数目一定为最大 $R[1]$。
- Generalization: 假设对于长度为 $m-1$ 的一条街，其抢劫后的最大数目为 $R[m-1]$，而对于长度为 m 的一条街，其抢劫后最大数目为 $R[m]$，故对于长度为 $m+1$ 的一条街，根据两种情况考虑：
  （1）如果第 m 家已经被抢劫，那么第 $m+1$ 家无法被抢劫，故 $R[m+1] = R[m]$
  (2) 如果第 m 家没有被抢劫，那么可以抢劫第 $m+1$ 家，故 $R[m] = R[m-1], R[m+1] = R[m] + T[m+1]$，故有

$$R(m+1) = \max\{R(m), R(m-1) + P(m+1)\} \tag{3}$$

得证

Analysis of Complexity

时间复杂度: 根据分析，实际上需要对从 1 到 n 的每个数进行一次计算和比较，故时间复杂度应该为: $O(n)$

空间复杂度: 由于新开了一个长度为 n 的数组，所以空间复杂度为: $O(n)$

Problem 2

Assignment Problem3

Solution

对于一个数字 n 来说，选择一个数字为 i(其中 $1 \leq i \leq n$)，设其为这棵 BST 的根节点，那么就将数据分成了两部分，其中前一部分数据为 $1-i-1$ 共 $(i-1$ 个数字) 和后一部分 $n-i$ 个数字。

算法思路: 根据上面的思路，对一个数字 n 的 BST 结构维他奶可以变成一些子问题，并迭代不同的划分策略，通过求和得到最终的结果。这里有一个特殊情况，也就是当左子树和右子树中没有节点的时候，其可能性为 1。

即有:

$$X(n) = \sum_{i=1}^{n} X(i) \times X(n-i) \tag{4}$$

Pseudo-Code

Pseudo-Code shown as follows.

Algorithm 2 BST Divide

Input: n ▷ n represents the given number.

```
Initialization: k, X                          ▷ k is an integer, X is a array with length n+1
Initialization: X[0] ← 1, X[1] ← 1
for i ← 2 to n do
    X[i] = 0
    for k ← 1 to i do
        X[i]+ = X[k − 1] × X[i − k]
    end for
end for
```

Output: X[n]

Proof of the correctness

使用结构归纳法证明如下:

- Initialization: 对于数字 1，其只有一种可能的 BST 结构，故此时 BST 可能的结构数量为 1，即 $X[1] = 1$（对于 $X[0]$，同样有 $X[0] = 1$)
- Generalization: 假设对于从 2 到 n 的数字，其对应划分为 BST 结构的方式分别为 $X[i], i \in 3, 4, ..., n$，那么对于一个数字 $n+1$,，选择一个数字 i 满足中 $1 \leq i \leq n+1$ 作为根节点，那么按照其可以进行划分为左子树和右子树（分别个数为 $i-1$ 和 $n+1-i$，故在这种情况下的可能 BST 结构为 $X[i-1] \times X[n+1-i]$，故对于所有的 i 选择进行加和计算，有此 $n+1$ 数的划分方式的计算应该为: $X[n+1] = X[0][n] + ... + X[i-1][n+1-i] ... + X[n][0]$。故一定能找到 $n+1$ 数的划分方式，得证。

Analysis of Complexity

时间复杂度根据伪代码，可以看到，有两层循环，每层循环的规模都是 n，所以时间复杂度为：$O(n^2)$
空间复杂度由于需要建立一个长度为 n 的数组，故空间复杂度为：$O(n)$

Problem 3

Assignment Problem4

Solution

对于一个长度为 n 的字符串 S 而言，需要将其划分为多个单词，要满足这样的要求，则存在一种可能的划分，使得其能在字典中找到对应的单词。

算法思路：假设对于从 $i-j$ 这一段字符串，可以划分为多个能够在字典中找到的单词，那么必定存在一个划分，使得 $S[i,k]$ $S[k,j]$，其中 $i \leq k \leq j$。假设判断函数为 f，根据上面的分析，有：

$$f(S[i,j]) = \sum_{k=i}^{j} \{f(S[i,k]) \ \& \ f(S[k,j])\} \tag{5}$$

Pseudo Code

故，根据上述分析，可以将对字符串 S 是否可以划分的判断定为 $f(S[0,n])$，具体可以用伪代码表示为
Pseudo-Code shown as follows.

Algorithm 3 String Match

Input: S, n, T ▷ S denotes the original string, and n is its length. T is the dictionary table.

```
function MATCH(S, X, i, j)
    if string S[i,j] is a word in T then
        return True
    end if
    for k ← i to j do
        X[i,j] ← (MATCH(X, S, i, k) & MATCH(X, S, k, j))
        if X[i, j] = True then
            return True
        end if
        return False
    end for
end function
Initialization: X                                   ▷ X is a two-dimensional array
X[0, n] = MATCH(X, S, 0, n)
```

Output: X[0,n]

Proof of the correctness

要证明的循环不变量为：判断长度为 n 的字符串 S 是否可以被字典划分，即判断是否存在一个划分将字符串划分为左右两部分（设为 $S[0,i]$ 和 $S[i,n]$ ），并且这两部分都能被字典划分为单词集合或该字符串本身就是字典中的一个单词。

使用结构归纳法证明如下：

- Initialization：此时 n 为 0，因此 $S[0,0]$ 和 $S[0,0]$ 均满足可划分的要求，而 $S[0,n]$ 也满足该要求，故满足循环不变量。
- Maintenance: 对于字符串 $S[i,j]$，首先需要判断当前该字符串是否能够被识别为一个单词，如果不能，那么意味着其要不不能被识别为单词集合，要不可以进行划分，在这两种情况下，通过判断所有可能的划分情况，来满足所设定的循环不变量。
- Termination: 终止条件为找到一个划分中，该字符串 $S[i,j]$ 满足其字典中的匹配，此时满足循环不变量。

Analysis of Complexity

时间复杂度根据计算逻辑，可以知道，实际上要计算一个二维数组中的所有值，并且对于每次计算是一个比较操作，假设比较实用的是 hash 方式，则时间复杂度为：$O(n^2)$，若使用的是查表方式，设表项数目为 m，则时间复杂度为 $o(mn^2)$

空间复杂度由于需要建立二维，每一维的长度为 n 的数组，故空间复杂度为：$O(n^2)$