算法设计与分析 LECTURE 10

第十一周作业-solution

LECTURER: 杨启哲 LAST MODIFIED: 2023 年 12 月 25 日

1. (教材习题 6.25) 为了降低背包问题运行的时间界限 $O(nW)$，我们尝试用这样的思路，我们使用一个大数 K 将容量 W 和每件物品的体积 w_i 都缩小 K 倍。具体来说，令 $W \leftarrow \lceil \frac{W}{K} \rceil$，$w_i \leftarrow \lceil \frac{w_i}{K} \rceil$；再在这个背包问题的新实例上运行我们提出的解决背包问题的算法。

   (1) 现在算法的时间复杂度是多少?

   (2) 这样子的策略其实并不能总得到原实例的最优解，请给出一个反例。

解答.

- 回顾一下背包问题的算法：

  背包问题

  输入: n 个物品，重量分别为 $w_1, \ldots, w_n$，价值分别为 $v_1, \ldots, v_n$，背包容量为 W
  输出: 最大价值

  for $j = 0$ to W do
      $V[j] \leftarrow 0$
  end for
  for $i = 1$ to n do
      for $j = W$ to w_i do
          if $j \geq w_i$ then
              $V[j] \leftarrow \max\{V[j], V[j - w_i] + v_i\}$
          end if
      end for
  end for
  return $V[W]$

  如果将 $W \leftarrow \lceil \frac{W}{K} \rceil$，$w_i \leftarrow \lceil \frac{w_i}{K} \rceil$，不难发现算法中的循环总共执行了 $\lceil nW/K \rceil$ 次，从而算法的时间复杂度为 $O(nW/K)$。

- 考虑如下的反例：

  - $W = 10$，物品重量为 $w_1 = 9, w_2 = 1$，价值分别为 $v_1 = 1, v_2 = 10$。

  如果 $K = 2$，则我们有：

  - $W = 5$，物品重量为 $w_1 = 5, w_2 = 1$，价值分别为 $v_1 = 1, v_2 = 10$。

  在原始问题中，最优解是两个物品全取，总价值为 11。但在更改后的问题中，最优解是只取第二个物品，总价值为 10。

□

Remark 0.1

需要注意的是,K 并不是一个常数，所以计算复杂性的时候不能像省略常数一样省略。

2. (教材习题 6.34) 请对旅行商问题设计一个动态规划算法，这里旅行商问题指的是：给定 n 个城市和一个 $n \times n$ 的距离矩阵 D，其中 $D[i,j]$ 表示从城市 i 到城市 j 的距离，旅行商要从某个城市出发，经过每个城市恰好一次，最后回到出发的城市，问旅行商的最短路线是什么？请给出算法的时间复杂度。

解答. 我们依旧用动态规划的思想来解决这个问题。需要注意的是，为了使经过的城市不重复，我们还需要记忆所有去过的城市。假定 1 是出发城市，我们令 $C[i][S]$ 为从城市 1 出发，经过 S 中的城市恰好一次，最后回到城市 i 的最短路线长度。显然 S 是一个包含 1 的子集。我们有如下递推关系：

$$C[i][S] = \begin{cases} 0 & \text{if } S = \{1\} \wedge i = 1 \\ \infty \text{if } |S| > 1 \wedge i = 1 \\ \min_{j \in S, j \neq i}\{C[j][S \setminus \{i\}] + D[j][i]\} & \text{otherwise} \end{cases}$$

其中 S 是一个集合，表示已经去过的城市。根据这个递推关系，我们可以给出如下的算法：

旅行商问题

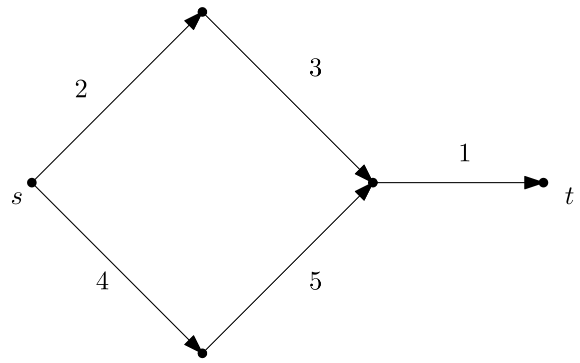
输入: 距离矩阵 D，城市数 n
输出: 最短路线长度

```
C[1][{1}] ← 0
for s = 2 to n do
    for 所有包含 1 大小为 s 的子集 S ⊆ {1, 2, ..., n} do
        C[1][S] = ∞
        for 所有 j ∈ S, j ≠ 1 do
            C[j][S] ← min_{i∈S,i≠j}{C[i][S \ {j}] + D[i][j]}
        end for
    end for
end for
return min_i{C[i][{1, 2, ..., n}] + D[i][1]}
```

注意到子集一共有 2^{n-1} 个，而每次更新 $C[j][S]$ 需要 $O(n^2)$ 的操作，因此整个算法的时间复杂度为 $O(n^2 2^n)$。 □

3. (教材习题 15.4) 证明或者否定下列结论：如果一个网络中所有的容量值是互不相同的，则存在一个唯一的流函数，其给出一个最大流。

解答. 这个结论很遗憾是不对的，考察下面的反例：



显然最大流是 1。但我们可以构造出两个不同的最大流：

- 从上方经过 1 的流
- 从下方经过 1 的流

□