

算法基础第五次作业

艾语晨 PB18000227

2020 年 11 月 17 日

目录

1	2
1.1 并查集	2
1.2 0-1 背包问题	2

第五次作业

第 1.1 题 并查集

Algorithm 1: MAKE-SET(x)

```
1 Let  $o$  be an object with three fields, next, value, and set
2 Let  $L$  be a linked list object with head = tail =  $o$   $o$ .next = NIL
3  $o$ .set =  $L$ 
4  $o$ .value =  $x$ 
5 return  $L$ 
```

Algorithm 2: FIND-SET(x)

```
1 return  $o$ .set.head.value
```

Algorithm 3: UNION(x, y)

```
1  $L1 = x$ .set
2  $L2 = y$ .set
3  $L1$ .tail.next =  $L2$ .head
4  $z = L2$ .head
5 while  $z$ .next  $\neq$  NIL do
6   |  $z$ .set =  $L1$ 
7  $L1$ .tail =  $L2$ .tail
8 return  $L1$ 
```

第 1.2 题 0-1 背包问题

假设我们知道解中有重量为 w 的特定项。然后，我们必须解决最大权重为 $W - w$ 的 $n - 1$ 个子问题。因此，要采取自下而上的方法，我们必须解决所有物品的 0-1 背包问题，并且可能

的重量小于 W 。我们将建立一个 $(n+1) \times (W+1)$ 值的表格，其中，行按项目编制索引，列按总重量编制索引。（表的第一行和第一列将是一个虚拟行）。对于第 i 行 j 列，我们通过比较包含项目 1 到 $i-1$ 且最大重量为 j 的背包的总价值以及包含项目的总价值，来决定将项目 i 包括在背包中是否有利。从 1 到 $i-1$ ，最大权重为 $j-i.weight$ ，还有项 i 。为了解决这个问题，我们只需要检查表的 n, W 项来确定可以达到的最大值。要阅读我们包含的项目，请从条目 n, W 开始。通常，请按照以下步骤操作：如果条目 i, j 等于条目 $i-1, j$ ，则不包括项目 i ，然后检查条目 $i-1, j$ 。如果条目 i, j 与条目 $i-1, j$ 不相等，则包括项 i ，然后检查条目 $i-1, j-i.weight$ 。

Algorithm 4: 0-1 Pack

Input: n, W

```

1 定义一个  $(n+1) \times (W+1)$  大小的矩阵  $K$ 
2 for  $j = 1$  to  $W$  do
3    $K[0][j] = 0$ 
4 for  $i = 1$  to  $n$  do
5    $K[i][0] = 0$ 
6 for  $i = 1$  to  $n$  do
7   for  $j = 1$  to  $W$  do
8     if  $j < i.weight$  then
9        $K[i][j] = K[i-1][j]$ 
10     $K[i, j] = \max(K[i-1][j], K[i-1][j-i.weight] + i.value)$ 
```
