HW 5

肖桐 PB18000037

2020 年 11 月 17 日

解 1.

```
S *MAKESET(int x)
{
    pSet = (S *)malloc(sizeof(S));  //pSet 指向一个新的集合头节点
    pNew = (Node *)malloc(sizeof(Node));    //pNew 指向一个新的数据节点
    pNew->Value = x;
    pNew->pNext = NULL;
    pNew->Set = pSet;   //指向头节点
    pSet->head = pNew;
    pSet->tail = pNew;
    pSet->Length = 1;
    return pSet;
}

S *FIND_SET(Node *x)
{
    return x->Set;
}

S *UNION(Node *p1, Node *p2)
{
    S *pSet1 = FIND_SET(p1);
    S *pSet2 = FIND_SET(p2);
    if (pSet2->Length > pSet1->Length)
    {
        Node *z = pSet1->head;
        while (z != NULL)
        {
            z->Set = pSet2;
            z = z->pNext;
        }
        pSet1->tail->pNext = pSet2->head;
        pSet2->head = pSet1->head;
        pSet2->Length += pSet1->Length;
```

```
        free(pSet1);
        return pSet2;
    }
    else
    {
        Node *z = pSet2->head;
        while (z != NULL)
        {
            z->Set = pSet1;
            z = z->pNext;
        }
        pSet2->tail->pNext = pSet1->head;
        pSet1->head = pSet2->head;
        pSet1->Length += pSet2->Length;
        free(pSet2);
        return pSet1;
    }
}
```

解 2. 设 $Value[i,j]$ 为选择前 i 个物品中的其中几个加入背包中，且使得总重量不超过 j 所取得的最大价值.
初始化 $Value[n+1, W+1]$
设 n 件物品重量为 $w_1, w_2, \ldots, w_n$，对应的价值分别为 $v_1, v_2, \ldots, v_n$.
则有最优子结构：$Value[i,j] = \max\{Value[i-1,j], Value[i-1, j-w_i] + v_i\}$.
理解为每次尝试将最新纳入考虑的第 i 件物品装入背包，若将第 i 件物品装入背包能够使得总价值增加的话，那就将第 i 件物品装入背包，否则保持原来的状态.
因此采取自底向上的方法可以写出伪代码：

```
for (int i = 0; i < n + 1; i++)
{//初始化
    Value[i, 0] = 0;
}
for (int j = 0; j < W + 1; j++)
{//初始化
    Value[0, j] = 0;
}

for (int i = 1; i < n + 1; i++)
{
    for (int j = 1; j < W + 1; j++)
    {
        Value[i, j] = max{Value[i - 1, j], Value[i - 1, j - w_i] + v_i};
    }
}
```

因为内循环复杂度为 $O(nW)$, 因此算法总复杂度为 $O(nW)$.