算法作业二参考答案

1. 最长等差子序列：给定一组整数数组和一个整数，不删除任何元素或删除数组中的元素而不改变其余元素顺序的序列称为子序列。请找出并返回数组中最长等差子序列的长度，该子序列中从左至右相邻元素之间的差等于。例如，数组，当 时最长的等差子序列是满足；当 时，则最长的等差子序列的长度1，即单个元素是最长等差子序列。令一维数组表示以作为末尾的最长等差子序列的长度。利用动态规划求解最长等差子序列，写出解题思路，动态规划方程。给出数组，的求解过程。

**参考答案：**

**答：令一维数组表示以作为结尾的最长等差子序列的长度。**

**有两种情况：**

**1）之前的元素，即最长等差子序列只有本身，即=1。**

**2）之前的某个元素满足，此时只需将添加到作为末尾的最长等差子序列，便可构成新的一个等差子序列，即=+1。**

**以作为末尾的等差子序列中，至少存在只包含一个元素的子序列，则其长度为1，即可以初始化。之前的元素逐一从小到大进行遍历，满足时便可获得以作为末尾的最长等差子序列的长度。根据以上分析，状态转移方程为：**

**最长等差子序列的长度。**

1. 给定本书和一个容量为的背包。每本书的重量为，价格为，。价格、容量和重量都是整数，价格单位为元，重量单位为50g。目标是让背包携带的书本的总价最大？动态规划求解问题。假设剩余容量为,从书本中选择装载了最优书本价值。

1）简单说明问题的最优子结构性质；

2）写出求解迭代递归公式；

3）用例子，和, 通过动态规划算法填下表1中缺少的数据，说明复杂度。

表1 表格

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | =0 | 1 | 2 | 3 | 4 |
| =0 |  |  |  |  |  |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |

答：1）该问题是典型的0-1背包问题，假设是问题本书的一个最优解，则是问题本书，其重量为的一个最优解。否则，存在是问题本书，其重量为的一个最优解，满足，则有，可得是问题本书的最优解，与假设矛盾。因此问题具有最优子结构性质。

2）初始条件 =0，无装载书本，背包书本价值为0元。初始条件 =0，装载容量为0，无书本，背包为0元。根据最优子结构性质，可以得到递归迭代式：

其中第1种情况为重量为的书本无法装载到背包里面；第2种情况为重量为的书本，对比加入或不加入状态，取二者中的最大的状态。

3）算法计算顺序从第1列的数值，然后再计算第2列的数值，以此类推，计算复杂度为O(nM)。

表1 表格

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | =0 | 1 | 2 | 3 | 4 |
| =0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 2 | 0 | 1 | 2 | 3 | 3 |
| 3 | 0 | 1 | 2 | 4 | 5 |

1. 给定一个以字符串表示的非负十进制整数*num*，移除这个数中的 *k* 位数字，使得剩下的数字的值最小，注意*num*的位数不小于*k*，且*num*不会包含任何前导零。例如，*num* = "1432219"，当*k*=3时，移除掉三个数字 4、3和 2 就可以组成一个新的最小的数字1219。请设计一个算法，
2. 写出求解该问题的**算法思路**。
3. 写出**核心伪代码**。
4. 分析**算法复杂度**。

参考答案：

算法思路：

两个相同位数的数字大小关系取决于第一个不同的数的大小。遍历字符串，如果发现前一个数字大于后一个数字，则将前一个数字删除，使得高位数字尽可能小。

int\* remove\_K(int str\_num[ ], unsigned int *k*)

//输入：字符数组str\_num（存储表示的非负十进制整数*num*）

//输入：非负整数*k*（移除这个数中的 *k* 位数字）

//输出：最小数字的字符数组

{

int s,i,len,flag=1;

int len = strlen(str\_num); //len是str\_num的长度，即num数的初始位数

while(s!=0) //只要s不是0，取数的工作就没有做完！

{

i = 0;

while(str\_num [i]<= str\_num [i+1]) //括号内的条件保证了不降序的条件，当它退出时，就是升序数列的末尾了

i++;

while(i<len-1) //这时已经找到了要取出的数str\_num [i]，以下是取出过程

{ str\_num [i]= str\_num [i+1]; i++;}

len--; //取出后数字长度减1

s--; //消耗掉一次取出次数

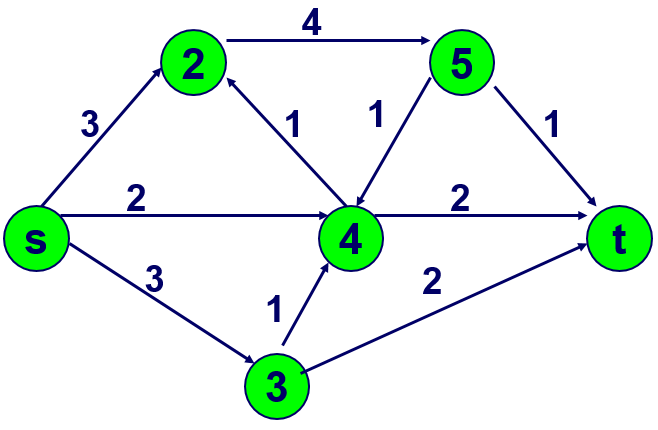
}// while(s!=0)

//输出时要小心最高位是0的问题！处理输出……

return str\_num;

}

1. 给定一个流网络G，如下图所示，源点*s*，汇点*t*，请问从*s*到*t*最多能输送多大的流？请*采用*Ford-Fulkerson*算法，并画图表示出该*问题**每一步的求解过程**（用BFS），只有答案者不得分。最后写出该网络的“最小割”的节点划分。



**参考答案：**

