算法作业 参考答案

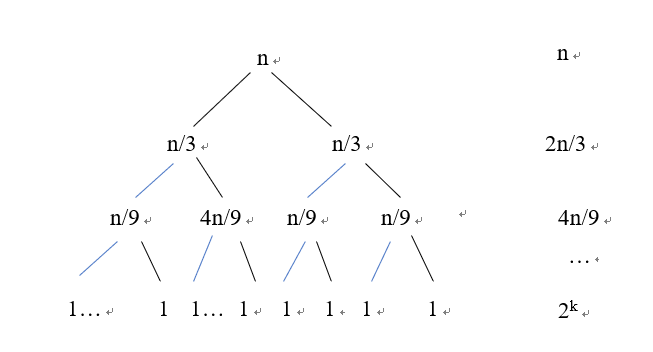
1、试解下面的递推关系：

（1）要使用**代入法**或**递归树**来求解，不能使用主定理。

（2）请使用**主定理**来求解*T*(*n*)的时间复杂性。

**参考答案：**

**1、递归树: n=3k**



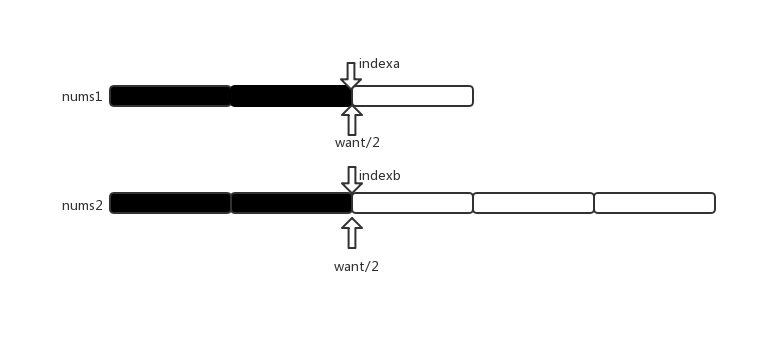
**2、主定理：**

2、假定两个长度为*m*，*n*的有序数组nums1和nums2，找出这两个有序数组合并后的中位数，要求时间复杂度为O(log(*m*+*n*))。（所谓中位数是指，当*n*为奇数时，处于中间位置的数值即为中位数；当*n*为偶数时，中位数则为处于中间位置的2个数的平均值。）请写出**算法的设计思路**，写出**核心伪代码，**并分析该算法的**时间复杂度**，要求有步骤。

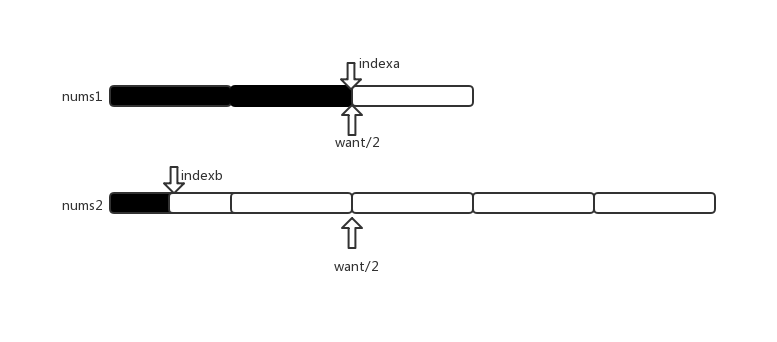
**参考答案：**

算法设计思路：

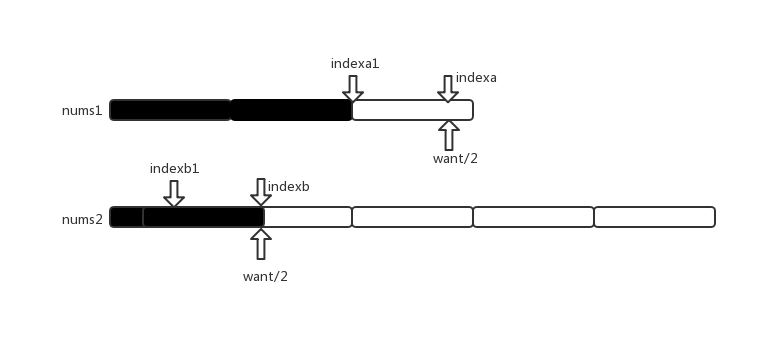
1. 需要求的中位数mid是把两个数组并集构成的数组num3的元素分为数目相等的两部分。设小于mid的元素构成集合C。
2. 计算两个数组元素总长度sumLen
3. 需要向C中添加的元素长度为(sumLen+1)/2 向下取整
4. 最开始仍需要添加到C中元素的长度为want=(sumLen+1)/2 向下取整，取索引值为indexa=indexb=want/2;



1. 比较nums1[indexa]与nums2[indexb]的大小(谁数值大，谁索引二分)，假设nums1[indexa]< nums2[indexb]，就对nums2再次进行二分找元素，直到nums1[indexa]>nums2[indexb]，假设indexb连续二分两次过后（indexb=indexb/2），nums1[indexa]>nums2[indexb]，然后把nums1索引z为indexa和indexa之前的indexa+1个元素和nums2索引为indexb和indexb之前的indexb+1个元素添加到数组C中，

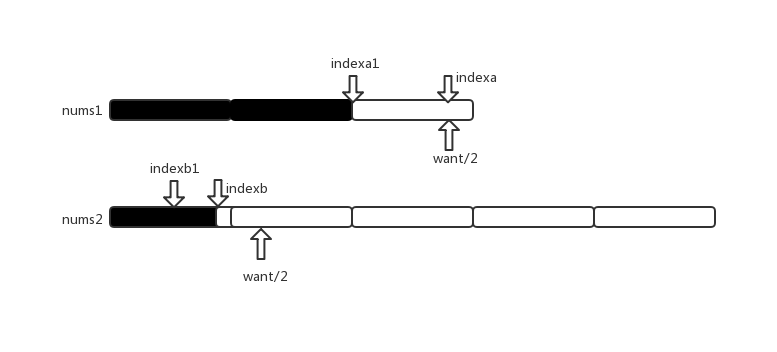


1. 更新want的值，计算want=(sumLen+1)/2-indexa-1-indexb-1。
2. 记录上一次indexa和indexb的值为，indexa1=indexa，indexb1=indexb
3. 更新indexa，indexb，indexa=indexa1+want/2，indexb=indexb+want/2



假设nums1[indexa]还是小于nums2[indexb]，就对nums2再次进行二分找元素,直到nums1[indexa]>nums2[indexb]，

假设indexb一次二分过后，nums1[indexa]>nums2[indexb]，然后把nums1索引为indexa1之后和indexa之前的元素（左开右闭）和nums2索引为indexb1之后和indexb之前的元素（左开右闭）添加到数组C中，



重复做下去，直到indexa+1+indexb+1 = sumLen/2

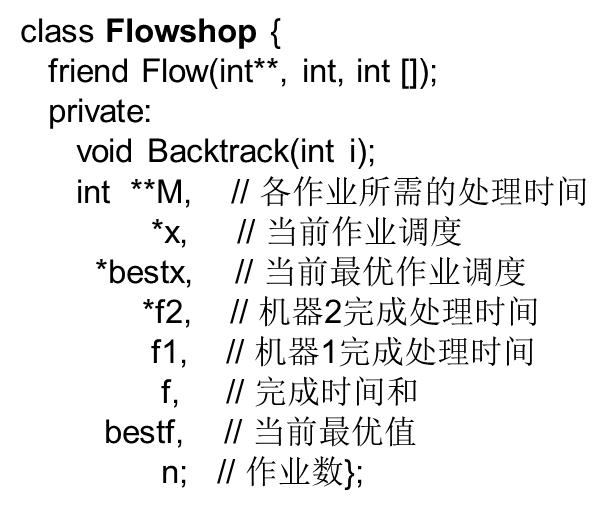
　　如果sumLen为奇数，说明nums[index+1]与nums2[indexb+1]两个元素较小的那个元素就是最终数组nums3的中位数,结束。

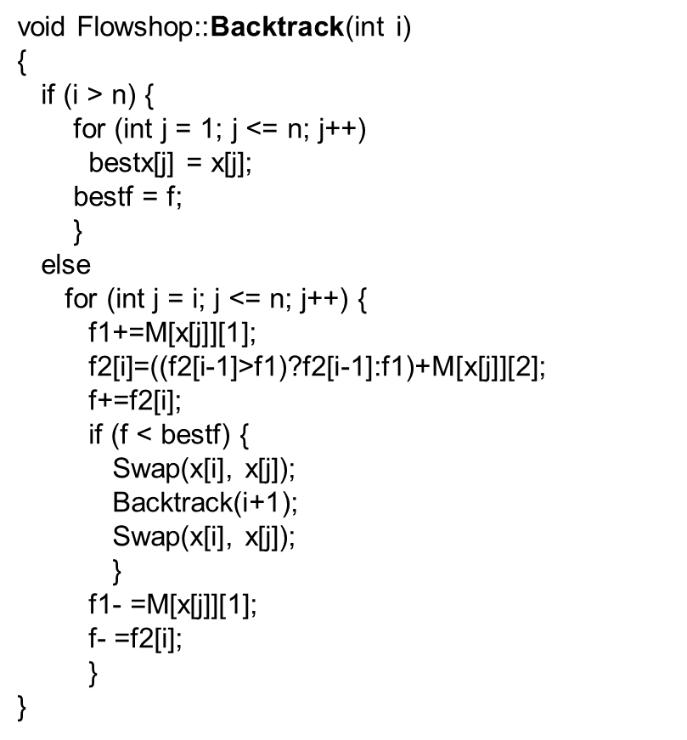
　　如果sumLen为偶数，记录nums[indexa]与nums[indexb]中较大的值为c1，记录nums[indexa+1]与nums[indexb+1]中较小的值为c2，最终数组nums3的中位数为(c1+c2)/2

3、给定*n*个作业的集合{*J*1, *J*2, …, *Jn*}。每个作业必须先由机器1处理，然后由机器2处理。作业*i*需要机器*j*的处理时间为*tji*。所有作业在机器2上完成处理的时间和称为该作业调度的完成时间和。批处理作业调度问题要求对于给定的*n*个作业，制定最佳作业调度方案，使其完成时间和达到最小。问题：用**回溯法**写出计算最优调度策略的**伪代码**，并**计算**下表所示的最优调度策略。

|  |  |  |
| --- | --- | --- |
| tji | 机器1 | 机器2 |
| 作业1 | 2 | 3 |
| 作业2 | 4 | 2 |
| 作业3 | 3 | 4 |

解：首先定义数据结构





（2）最优调度策略为1->3->2，耗时11.

1. 求三个字符串的X、Y、Z的最长公共子序列LCS（X,Y,Z）
2. 写出算法思想与递推式。
3. 写出核心伪代码。
4. 分析算法复杂度。

**参考答案：**

算法思想：

dp[i][j][k]表示A[0...i],B[0...j],C[0...k]的LCS

递推式：

if (a[i]==b[j]&&b[j]==c[k])  
dp[i][j][k]=dp[i-1][j-1][k-1]+1;  
else  
dp[i][j][k]=max(max(dp[i][j][k], dp[i-1][j][k]), max(dp[i][j-1][k], dp[i][j][k-1]));

伪代码如下：

function LCS(A,B,C)

 n1 = A.length;

 n2 =B.length

n3 =C.length

//字符串的个数可以是0-n,共n+1个可能，所以循环中是i<=n

for i=0 to n1

 for j=0 to n2

   for k=0 to n3

     //只要有一个序列的长度为0，公共子序列的长度为0

     if i==0||j==0||k==0 then a[i][j][k] = 0;

 //自底向上求a[i][j][k]

for i=1 to n1

 for j=1 to n2

   for k=1 to n3

      {

         if A[i - 1] == B[j - 1] && B[j - 1] ==C[k - 1])

         then  a[i][j][k] = a[i - 1][j - 1][k - 1] + 1;

        else   a[i][j][k] = max(max(a[i - 1][j][k], a[i][j - 1][k]),a[i][j][k -1]);

      }

return a[n1][n2][n3]

时间复杂度O(n3)

5、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，

1. 写出求解该问题的**算法思路**。
2. 写出**核心伪代码**。
3. 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;

}

6、在某国海域上有6座孤岛，其之间的距离（单位km）如下表所示。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F |
| A | 0 | 3.2 | 4 | 1.5 | 3 | 5 |
| B | 3.2 | 0 | 2.6 | 5.9 | 1.7 | 2.3 |
| C | 4 | 2.6 | 0 | 4.9 | 3.3 | 2.9 |
| D | 1.5 | 5.9 | 4.9 | 0 | 5.5 | 2 |
| E | 3 | 1.7 | 3.3 | 5.5 | 0 | 4.1 |
| F | 5 | 2.3 | 2.9 | 2 | 4.1 | 0 |

现该国政府规划这6座孤岛间建桥，要使得所有孤岛连通，并且规划出造桥路径最短。现在请设计一个算法，写出**算法设计思路，**并画图表示出算法**每步求解过程**，只有答案者不得分。

**答案：**

**1）Kruskal算法可以求解最小生成树，该问题中造桥的代价就是权重，其本质上是利用权重最小的边连接所有孤岛，是一个求解最小生成树的问题，所以可以利用Kruskal算法求解。**

**2）Step1:**

**Step2:**

1.5

**…**

**结果**

1.5

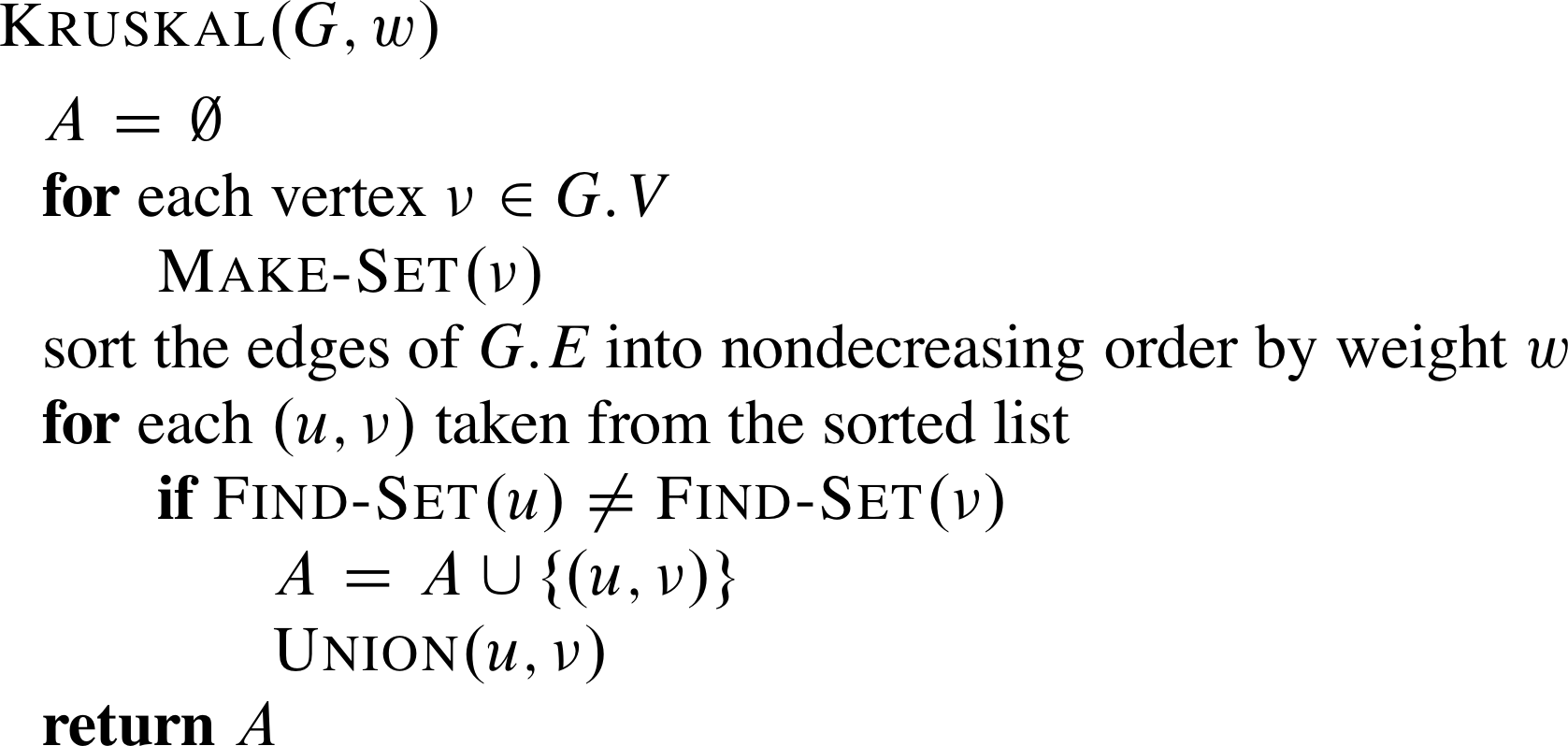
2.6

2

1.7

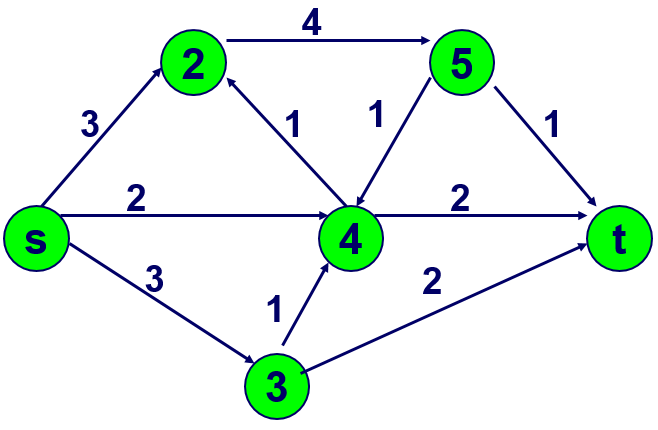
2.3

**3）算法 Kruskal(G)**

****

**时间复杂度：O(|E| lg |E|) 或者O(|E| lg |V |)**

7、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？请设计一个算法，写出**算法设计思路，**并画图表示出该问题**每一步的求解过程**，只有答案者不得分。



**参考答案：**

