算法作业一参考答案

1. 填空题：
   1. 时间，空间
   2. 递归树法，代入法
   3. **NP-hard 问题至少与 NP 中最难的问题一样难（即所有 NP 问题可在多项式时间归约到 NP-hard 问题）**
   4. 3。（残留容量 = 反向边容量 + 正向流量 = 0 + 3）
   5. **原图中每条边（u,v）对应两条残留边（u,v）和（v,u），剩余容量和反向流量**‌
   6. 否
   7. 否，是
   8. 是，否
   9. 树边，前向边，反向边，交叉边，反向边
   10. Θ(|*V*| + |*E*|) （需遍历所有顶点和边）
   11. **集合覆盖问题**‌
2. 使用递归树求解下面的递推式的时间复杂性。

（1）已知，计算.

（2）已知*T*(1) = 1，*T*(*n*) = 3*T*(*n*/3) + 2*T*(*n*/2) + *n*, *n* > 1；计算*T*(*n*)。

**参考答案：**

（1）**递归树:**

cn

n

cn/2

cn/2

cn

cn/4

cn

cn/4

cn/4

cn/4

cna

…

a…

a…

a

a1

a

a

a

a



（2）对于递推式 *T*(*n*) = 3*T*(*n*/3) + 2*T*(*n*/2) + *n*，可以构建如下的递归树：

根节点：代价为*n*

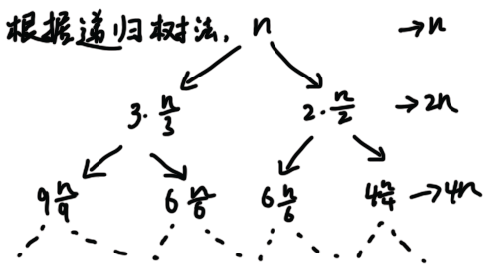
第1层：3个*T*(*n*/3)节点和2个*T*(*n*/2)节点，每个*T*(*n*/3)节点的代价为*n*/3，每个*T*(*n*/2)节点的代价为*n*/2，第一层总代价为 *n* + *n* = 2*n*

第2层：每个*T*(*n*/3)节点又会生成3个*T*(*n*/9)节点和2个*T*(*n*/6)节点，每个T(*n*/2)节点会生成3个*T* (*n*/6)节点和2个*T*(*n*/4)节点，这一层的总代价为2 + 2*n* = 4*n*

以此类推，直到达到基本情况 T (1) = 1

第*k*层：第*k*层总代价为2*kn*（通过前3层归纳：20*n* = *n*, 21*n* = 2*n*, 22*n* = 4*n*）。

最终总代价由最深层（log2​*n*）决定



递归树总代价求和

E:\Tencent\Weixin\WeChat Files\wxid_99zdtb01qt0321\FileStorage\Temp\1750478536402.png

1. 用主定理方法分析下列T(n)的时间复杂性。

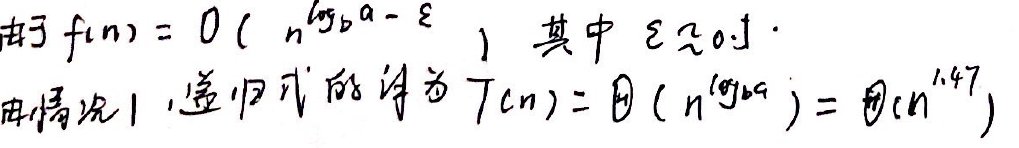
(1)

(2)

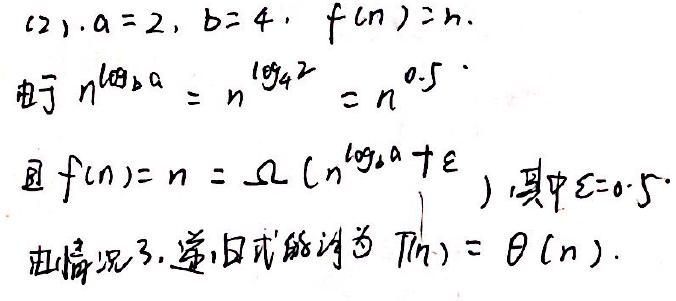
1. a=5; b=3; f(n)=nlogn







(2)



1. 给定一个由*n*个互不相同的数所组成的集合S。请利用分治思想设计一个算法，找出S的中位数。（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。）分析该算法的时间复杂度。

**参考答案：**

思路：利用快速排序算法实现。任意挑一个元素，以改元素为支点，划分集合为两部分，如果左侧集合长度恰为 (n-1)/2，那么支点恰为中位数。如果左侧长度<(n-1)/2, 那么中位点在右侧，反之，中位数在左侧。进入相应的一侧继续寻找中位点。

Mid(S,n)

输入：数组S， 数组中元素个数n

输出：数组的中位数

if n为偶数 then return Midk(S,n,n/2)

else return (Midk(S,n,n-1/2)+ Midk(S,n,n+1/2))/2

Midk(S，n, k)

输入：数组S， 数组中元素个数n，寻找k位数

输出：数组按顺序排列之后的第k位数

int L[n],R[n]

lp=0,rp=0

for i=1 to n-1

if S[i]> S[0] then R[rp++]=S[i]

else L[lp++]=S[i]

if(lp==k) then return S[0]

if(lp>(n-1)/2) then return Mid(L, lp，k)

else return Mid(R,rp，k-lp)

该算法的主要时间分析Midk(S，n, k)

首先进行左右子集的划分需要n-1次比较

然后递归调用左或者右，一般情况下，会剩下一半左右的子集

所以这个查找效率 T(n)=T(n/2)+n-1

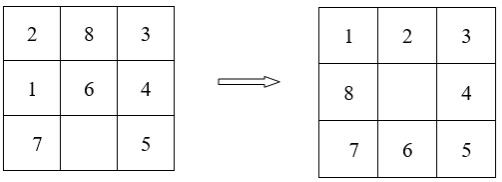
根据主定理，该算法的时间复杂度为O(n)

在最坏情况下，每次只能去掉1个元素，算法的效率变为

T(n)=T(n-1)+n-1

算法的时间复杂度变为O(n2)

1. 在3×3的棋盘上，摆有8个棋子，每个棋子上标有1~8的某个数字，棋盘中留有一个空格，空格周边的棋子可以移动空格中（左移、右移、上移和下移这四个操作）。初始状态和目标状态如下图所示，请使用回溯法来求解出本问题的所有解。



**参考答案：**

