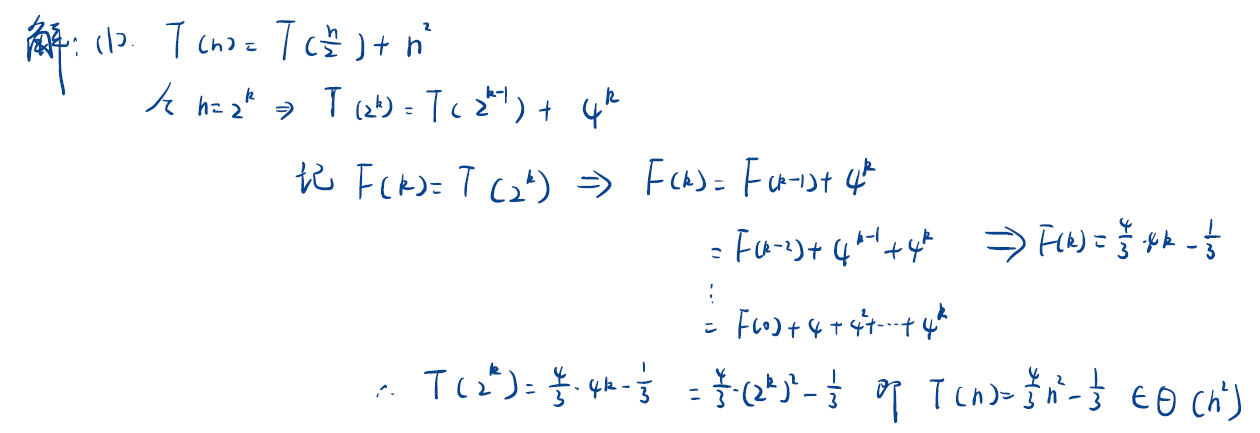
算法作业一参考答案

1. 试解下面的递推关系计算时间复杂性，要求不能使用主定理。

（1）已知 ，计算.

（2）已知，计算.

**参考答案：**



（2）

**递归树:**

cn

cn

cn/2

cn/2

cn

cn/4

cn

cn/4

cn/4

cn/4

cna

…

a…

a…

a

a1

a

a

a

a



1. 用主定理方法分析下列T(n)的时间复杂性。

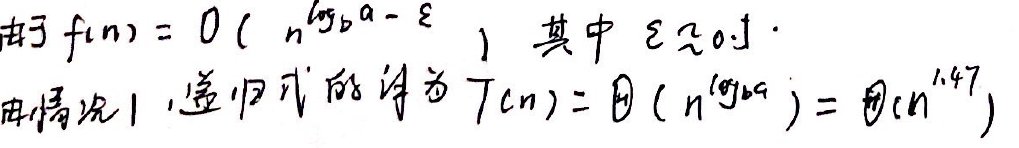
(1)

(2)

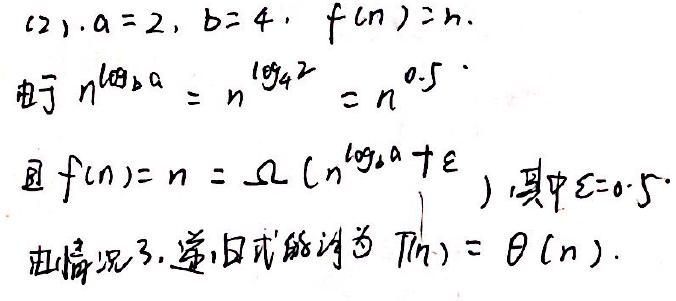
1. a=5; b=3; f(n)=nlogn







(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 Midk(L, lp，k)

else return Midk(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. （1）现在有一个卡车运送一批货物，货物重量如下表所示。卡车司机运送货物的收益包含两部分：（1）小于等于2吨每装载一件物品收益1000元，（2）重量大于2吨的物品按照每吨500元收取费用。请用分支界限算法设计写出相关程序伪代码，使得运费收益最大化。(2)如果物品重量如下表所示，卡车载重为5吨，应该装入哪些物品使得收益最大。

|  |  |
| --- | --- |
| **物品编号** | **重量(吨)** |
| 1 | 1 |
| 2 | 3 |
| 3 | 2 |
| 4 | 4 |

