**《算法设计与分析基础》（第一套）参考答案与评分标准**

|  |  |
| --- | --- |
| **适用专业年级（方向）：** | **计算机科学与技术2013级** |

**一、单项选择题（每题2分，共20分）**

1~5：C B A B C 6~10：A C B A C

二、**（10分）**

解：a. 设C(n)表示算法所做元素比较次数，则可得递推关系表达式：

C(n)=C(n-1)+2，n＞1 （1分）

C(1)=0 （1分）

b. 迭代求解：

C(n)=C(n-1)+2

=C(n-2)+2+2

=C(n-3)+2+2+2

=…

=C(n-i)+i\*2

=…

=C(1)+2(n-1)=2(n-1) （4分）

**评分标准：计算正确得4分，部分正确，酌情扣分。**

c. 由b的计算结果可知算法基本操作的时间效率类型为：Θ(n)。 （2分）

d. 二者的乘法基本操作次数完全一样，但是递归算法有额外的栈结构管理开销，非递归算法则没有，所以非递归算法更为高效。 （2分）

**三、（10分）**

解：a. 减治法中的减可变规模 （2分）

1. K=4 k-1=3

**15，**18，11，23，26，36，14

↑i ↑j 18与14互换

**15，**14，11，23，26，36，18

↑j ↑i

11，14，**15，**23，26，36，18 j=2<k-1 left←3

**23**，26，36，18 18与26互换

↑i ↑j

**23**，18，36，26

↑j ↑i

18，**23**，36，26 j=4>3 right←3

18 j=3=k-1 18即中位数

**评分标准：步骤正确得6分，结果正确得2分；部分正确，酌情扣分。**

四、**（26分）**

解：a. Algorithm ClosestNu(A[1..n])

//输入：一个长度为n的有序数组A

//输出：A中两个最接近元素的差值

dmin←A[2]-A[1]

for i←2 to n-1 do

temp←A[i+1]-A[i]

if temp<dmin dmin←temp

return dmin

**评分标准：算法思路与功能等价得10分；部分正确，酌情扣分。**

b. Algorithm ClosestDNu(A[l..r])

//输入：一个有序数组A，其起始下标为l，终止下标为r

//输出：A中两个最接近元素的差值

If l=r return ∞

else if l+1=r return A[r]-A[l]

else

mid←(l+r)/2

dmin1←ClosestDNu(A[l..mid])

dmin2←ClosestDNu(A[mid+1..r])

dmin3←A[mid+1]-A[mid]

if dmin1>dmin2 dmin1←dmin2

if dmin1>dmin3 dmin1←dmin3

dmin←dmin1

return dmin

**评分标准：算法思路与功能等价得10分；部分正确，酌情扣分。**

c. 假设直接求解算法中基本操作比较的执行次数为M(n)，则有

M(n)==n-2 所以其效率类型为：θ(n) （2分）

假设分治算法中基本操作比较的执行次数为M(n)，则有

S(n)=2S(n/2)+2 n>2

S(2)=0 S(1)=0

对以上递推式进行迭代展开求解得S(n)=n-2

所以其效率类型为θ(n) （2分）

（也可用减法作为基本操作，基本操作次数具体值也可随算法实现细节不同而不同。）

二者的效率类型相同，但是分治递归算法存在额外栈管理，所以直接求解算法更好。（2分）

五、**（14分）**

解：a、计算过程：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 系数 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| x=2 | 1 | 2\*1+1=3 | 2\*3+0=6 | 2\*6+1=13 | 2\*13+0=26 | 2\*26+0=52 | 2\*52+1=105 |

计算结果为105。

**评分标准：计算过程正确得6分，结果正确得2分；部分正确，酌情扣分。**

b、霍纳法则效率类型：Θ(n) （2分）

二进制幂算法效率类型：Θ(logn) （2分）

二进制幂算法的效率更好 （2分）

六、**（10分）**

解：

a. 主表 根表

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 |  |  |  |  |  | 0 | 1 | 2 | 3 | 4 |
| 1 | 0 | 0.2 | 0.8 | 1 | 1.7 |  |  |  |  | 1 |  | 1 | 2 | 2 | 2 |
| 2 |  | 0 | 0.4 | 0.6 | 1.3 |  |  |  |  | 2 |  |  | 2 | 2 | 2 |
| 3 |  |  | 0 | 0.1 | 0.5 |  |  |  |  | 3 |  |  |  | 3 | 4 |
| 4 |  |  |  | 0 | 0.3 |  |  |  |  | 4 |  |  |  |  | 4 |
| 5 |  |  |  |  | 0 |  |  |  |  | 5 |  |  |  |  |  |

具体计算过程：

C(1,3)=min k=1: C(1,0)+C(2,3)+S=0.6+0.7=1.3

k=2: C(1,1)+C(3,3)+S=0.2+0.1+0.7=1 √

k=3: C(1,2)+C(4,3)+S=0.8+0.7=1.5

C(2,4)=min k=2: C(2,1)+C(3,4)+S=0.5+0.8=1.3 √

k=3: C(2,2)+C(4,4)+S=0.4+0.3+0.8=1.5

k=4: C(2,3)+C(5,4)+S=0.6+0.8=1.4

C(1,4)=min k=1: C(1,0)+C(2,4)+S=1.3+1=2.3

k=2: C(1,1)+C(3,4)+S=0.2+0.5+1=1.7 √

k=3: C(1,2)+C(4,4)+S=0.8+0.3+1=2.1

k=4: C(1,3)+C(5,4)+S=1+1=2

**评分标准：计算过程及结果正确得8分；部分正确，酌情扣分。**

b. 最优二叉查找树为：（2分）

B

D

A

C

**七、（10分）**

解：a. 价值上界ub＝当前包内总价值＋剩余物品采用贪心策略连续装包

描述求解过程的状态空间树如下：

0

0

1

1

1

1

1

1

w=0, v=0

ub

=100

w=0, v=0

ub

=78

w=4, v=40

ub

=76

w=6, v=42

ub

=78

w=11

**不可行**

w=17

**不可行**

0

0

w=4, v=40

ub

=70

w=6, v=42

ub

=60

w=9, v=65

ub

=69

w=14, v=66

ub

=66

w=4, v=40

ub

=64

w=6, v=42

ub

=42

**最优解**

**不可能更优**

0

0

w=0, v=0

ub

=60

w=0, v=0

ub

=56

**不可能更优**

**评分标准：状态空间树正确得8分，部分正确酌情扣分。**

b.最优装包方案为物品1和物品3装包，能带走的最大价值为66。（2分）