郑源泽19307130077hw2

Homework on Chap4

- 4.2-3,4.3-4
- Solve the recurrent $T(n) = T(\sqrt{n}) + \theta(n)$
- A[1, 2, …,n] is an increasing-ordered array with all distinct integers (possibly be negative). Find an algorithm by Divide-and Conquer to find *i* that A[i]=i. The worst-case running time of your algorithm should be in O(lgn).

Homework of Chapter 5

- 5.2-4, 5.3-4
- Problem 5-1

4.2-3

> How would you modify Strassen's algorithm to multiply n*n matrices in which n is not an exact power of 2? Show that the resulting algorithm runs in time $\theta(n^{lg7})$

将 $n*n$ 的矩阵用0填充，使其维数成为大于n的最小的2的整数倍。假设原始矩阵算式为
$A*B=C$

经扩展得到A1，B1, 可以应用Strassen算法，得到 C1, 此时我们可以从C1中得到正确的答案C

$$A1=\begin{bmatrix}A & 0\\ 0 & 0\end{bmatrix}\quad B1=\begin{bmatrix}B & 0\\ 0 & 0\end{bmatrix}\quad C1=\begin{bmatrix}C & 0\\ 0 & 0\end{bmatrix}$$

时间复杂度：$\theta(m^{lg7})$, m是最小的大于n的2的整数次幂，故m<2n

$\theta(m^{lg7}) < \theta((2n)^{lg7}) = \theta(2^{lg7}\, n^{lg7}) = \theta(n^{lg7})$

4.3-4

> Show that by making a different inductive hypothesis, we can overcome the difficulty with the boundary condition T(1) = 1 for recurrence (4.19) without adjusting the boundary conditions for the inductive proof.

$$T(n)=2T(\lfloor n/2\rfloor)+n$$

猜测：$T(n) \leq nlgn + n$
代入：
$T(n) \leq 2(c\lfloor n/2 \rfloor lg\lfloor n/2 \rfloor + \lfloor n/2 \rfloor) + n \leq 2c(n/2)lg(n/2) + 2(n/2) + n = cnlg(n) + (2-c)n$

当 $c \geq 1$ 时，满足 $T(n) \leq cnlgn + n$

$T(1) = 1 \leq cnlgn + n = 0 + 1 = 1.$

求解递归式 $T(n) = T(\sqrt{n}) + \theta(n)$

改变变量，令 $m = log(n)$ 得到 $T(2^m) = T(2^{\frac{m}{2}}) + \theta(2^m)$

重命名 $S(m) = T(2^m)$ 得到 $S(m) = S(m/2) + \theta(2^m)$

应用主定理，a=1，b=2，f(m)=θ(2^m)

$log_b a$ = 0 $n^{log_b^a} = 1$ $f(m) = \Omega(m^{logb^{a+\delta}})$

$$f(\frac{m}{2}) <= cf(m), 2^{m/2} <= c * 2^m, c = \frac{1}{2}, 显然，当 m >= 2 时成立$$

由情况3，递归式的解为 $\theta(2^m)$ 从S转换回T，$T(n) = T(2^m) = S(m) = \theta(2^m) = \theta(n)$

算法如下

与二分搜索原理类似。由于题目中明确了每个数字都是不重复的，那么当我们发现某个数字大于他的索引，这表明其右侧的所有数字都会大于其索引，于是满足条件的数字一定在左侧。小于的情况同理。

T(n) = T(n/2) + c

最坏情况数字出现在边缘，O(lgn)

```
public class HomeWork2_3 {
    public static void main(String[] args) {
        // index :  0 1 2 3 4 5 6 7 8 9
        // value : -1 0 1 2 3 4 6 8 9 10
        int[] array = {-1, 0, 1, 2, 3, 4, 6, 8, 9, 10};
        Solution solution = new Solution();
        solution.find(array, 0, array.length - 1);
    }
}
class Solution{
    public void find(int[] arr, int left, int right){
```

```java
        if(left <= right){
            int mid = left + (right - left) / 2;
            if(arr[mid] == mid){
                System.out.println(mid);
            }
            else if(arr[mid] > mid){
                right = mid - 1;
                find(arr, left, right);
            }
            else{
                left = mid + 1;
                find(arr, left, right);
            }
        }
    }
}
```

第五章作业老师上课说可以下周一起交。
