第1次作业

1：抄写MERGESORT、MERGE

1.2-2

Suppose we are comparing implementations of insertion sort and merge sort on the same machine. For inputs of size n, insertion sort runs in $8n^2$ steps, while merge sort runs in $64n \lg n$ steps. For which values of n does insertion sort beat merge sort?

2≤n≤43

因为：lg43=5.426，43≤8lg43=43.408

lg44=5.459，44≥8lg44=43.672

这里lg函数以2为底

1.2-3

What is the smallest value of n such that an algorithm whose running time is $100n^2$ runs faster than an algorithm whose running time is 2^n on the same machine?

n≥15

- 思考题：2.2-2 选择算法

1）程序：

```
void SelectSort(RecordType r[], int length)    /*对记录数组r做简单选择排序，length为待排序记录的个数*/
{
    int temp;
    for ( i=0 ; i< length-1 ; i++) //n-1趟排序
    {
        int index=i；      //假设index处对应的数组元素是最小的
        for (int j=i+1 ; j < length ; j++)     //查找最小记录的位置
            if (r[j].key < r[index].key )
                index=j;
        if ( index!=i)
        {
            temp     = r[i];
            r[i]     = r[index];
            r[index] = temp;
        }
    }
}
```

2）循环不变式：子数组A[1~j-1]已按序排列，且A[1~j-1]中的元素均来自A[1~n]。

3）最优一个元素已经就位。

4）最好、最坏、平均：$\Theta(n^2)$

2-4 （逆序对） 假设$A[1..n]$是一个有 n 个不同数的数组。若 $i<j$ 且 $A[i]>A[j]$，则对偶(i,j)称为 A 的一个**逆序对**(inversion)。

a. 列出数组〈2，3，8，6，1〉的 5 个逆序对。

b. 由集合$\{1, 2, \cdots, n\}$中的元素构成的什么数组具有最多的逆序对？它有多少逆序对？

c. 插入排序的运行时间与输入数组中逆序对的数量之间是什么关系？证明你的回答。

d. 给出一个确定在 n 个元素的任何排列中逆序对数量的算法，最坏情况需要 $\Theta(n \lg n)$时间。（提示：修改归并排序。）

a. The inversions are $(1,5), (2,5), (3,4), (3,5), (4,5)$.

b．序列（n，n-1, ⋯, 2, 1）有最多的逆序对。

该序列的逆序对有 $\binom{n}{2} = n(n-1)/2$

c. 插入排序

```
INSERTION-SORT(A)
for j = 2 to A.length
    key = A[j]
    // Insert A[j] into the sorted sequence A[1..j − 1].
    i = j − 1
    while i > 0 and A[i] > key
        A[i+1] = A[i]
        i = i − 1
    A[i+1] = key
```

while循环里的每次交换都是因为key（在A[j]位置）与A[i]是逆序对，通过交换消除逆序对，所以插入排序的运行时间正比于输入数组中的逆序对的数量。

d. 类似MergeSort，在Merge的过程中，左侧子序列中任何大于右侧某元素的元素，都形成一个逆序对。所以对n个元素而言，逆序对的个数等于分治后左子序列中的逆序对个数+右子序列中逆序对个数，再加右子序列元素相对于左子序列元素形成的逆序对的个数。

COUNT-INVERSIONS(A, p, r)

$\quad$ $inversions = 0$

$\quad$ **if** $p < r$

$\quad\quad$ $q = \lfloor (p + r)/2 \rfloor$

$\quad\quad$ $inversions = inversions + \text{COUNT-INVERSIONS}(A, p, q)$

$\quad\quad$ $inversions = inversions + \text{COUNT-INVERSIONS}(A, q + 1, r)$

$\quad\quad$ $inversions = inversions + \text{MERGE-INVERSIONS}(A, p, q, r)$

$\quad$ **return** $inversions$

```
MERGE-INVERSIONS(A, p, q, r)
  n1 = q - p + 1
  n2 = r - q
  let L[1 .. n1 + 1] and R[1 .. n2 + 1] be new arrays
  for i = 1 to n1
      L[i] = A[p + i - 1]
  for j = 1 to n2
      R[j] = A[q + j]
  L[n1 + 1] = ∞
  R[n2 + 1] = ∞
  i = 1
  j = 1
  inversions = 0
  for k = p to r
      if R[j] < L[i]
          inversions = inversions + n1 - i + 1
          A[k] = R[j]
          j = j + 1
      else A[k] = L[i]
          i = i + 1
  return inversions
```

3.1-2 证明：对任意实常量 a 和 b，其中 $b>0$，有 $(n+a)^b = \Theta(n^b)$

To show that $(n+a)^b = \Theta(n^b)$, we want to find constants $c_1, c_2, n_0 > 0$ such that $0 \le c_1 n^b \le (n+a)^b \le c_2 n^b$ for all $n \ge n_0$.

Note that

$$\begin{aligned} n+a &\le n+|a| \\ &\le 2n \qquad \text{when } |a| \le n \ , \end{aligned}$$

and

$$\begin{aligned} n+a &\ge n-|a| \\ &\ge \frac{1}{2}n \qquad \text{when } |a| \le \tfrac{1}{2}n \ . \end{aligned}$$

Thus, when $n \ge 2|a|$,

$$0 \le \frac{1}{2}n \le n+a \le 2n \ .$$

Since $b > 0$, the inequality still holds when all parts are raised to the power b:

$$0 \le \left(\frac{1}{2}n\right)^b \le (n+a)^b \le (2n)^b \ ,$$

$$0 \le \left(\frac{1}{2}\right)^b n^b \le (n+a)^b \le 2^b n^b \ .$$

Thus, $c_1 = (1/2)^b$, $c_2 = 2^b$, and $n_0 = 2|a|$ satisfy the definition.

3.1-5　证明定理 3.1。

参考Ω、O、Θ的定义，直接证明即可：

Ω：如果存在两个正常数c和n_0，对于所有的$n \geq n_0$，有

$$|f(n)| \geqslant c|g(n)|,$$

则记作$f(n)=\Omega(g(n))$。

O：如果存在两个正常数c和n_0，对于所有的$n \geq n_0$，有

$$|f(n)| \leqslant c|g(n)|,$$

则记作 $f(n)=O(g(n))$。

Ω：如果存在正常数c_1，c_2和n_0，对于所有的$n \geq n_0$，有

$$c_1|g(n)| \leqslant |f(n)| \leqslant c_2|g(n)|,$$

则记作$f(n) = \Theta(g(n))$

需要包括：充分性证明、必要性证明（略）