Algorithm HW3

2.

```
随机整数数组排序耗时统计（单位：ms；每个规模平均 5 次）

n          MergeSort(ms)      QuickSort(ms)      Q/M        更快
----------------------------------------------------------------------
10000      1.303              1.050              0.806      Quick
30000      3.900              3.235              0.830      Quick
50000      6.753              5.618              0.832      Quick
80000      11.202             9.236              0.825      Quick
100000     14.539             11.808             0.812      Quick
200000     30.495             24.915             0.817      Quick

结论（基于本机本次随机数据与实现）：
在这组实验里 QuickSort 大多数规模更快（6/6 个规模胜出），平均 Q/M=0.820（小于 1 表示 QuickSort 更快）
原因通常是：两者平均时间复杂度都近似 O(n log n)，但 QuickSort 常数因子更小、缓存友好；
而 MergeSort 需要额外 O(n) 辅助数组并进行大量拷贝/合并操作。
```

3. 设待排序数组规模为 n。MergeSort 在递归过程中，每一层递归都会在“合并阶段”使用一个长度为当前子数组规模的辅助数组；但这些辅助数组不会在同一时间叠加，而是随着递归返回逐层释放。递归深度为 $\log n$，但任意时刻额外占用的最大辅助空间来自最顶层合并，为 n。递归调用栈的空间复杂度为 $O(\log n)$，远小于辅助数组。

4. 设

$$R(n) = \max_k C_A^k(n)$$

由题给递推式：

$$C_A^k(n) \le cn + \frac{1}{n}\left(\sum_{i\le k} C_A^{k-i}(n-i) + \sum_{k<i\le n} C_A^k(i-1)\right)$$

对两项取上界 $R(\cdot)$，得：

$$R(n) \le cn + \frac{1}{n}\left(\sum_{i=1}^{k} R(n-i) + \sum_{i=k+1}^{n} R(i-1)\right)$$

注意右侧两项合并后覆盖了 $R(1)$ 到 $R(n-1)$ 的所有项，因此：

$$R(n) \le cn + \frac{1}{n}\sum_{j=1}^{n-1} R(j)$$

归纳假设： 对所有 $j < n$，有 $R(j) \le 4cj$。代入得：

$$R(n) \le cn + \frac{1}{n}\sum_{j=1}^{n-1} 4cj = cn + \frac{4c}{n} \cdot \frac{(n-1)n}{2}$$

$$R(n) \le cn + 2c(n-1) = 3cn - 2c < 4cn$$

归纳成立。