学号：**ZY2206117**　姓名：**黄海浪**　作业：**homework1**

1. 题目 1

设

$$F(n) = C(n) - 1$$

当$n \geq 2$时，

$$F(n) = 2\left(F\left(\frac{n}{2}\right) + 1\right) + n - 2$$
$$= 2F\left(\frac{n}{2}\right) + n$$

所以有

$$F(n) = \begin{cases} 0, & n = 1 \\ 2F\left(\frac{n}{2}\right) + n, & n \geq 2 \end{cases}$$

利用定理 1，可将$a = 2$，$b = 1$，$c = 2$，$d = 0$，$x = 1$带入得

$$F(n) = nlog_2 n + 0 = nlog_2 n$$

所有有

$$C(n) = nlog_2 n + 1$$

所以$C(n)$的渐进复杂性为$O(nlog_2 n)$

2. 题目 2

（1）

给定一张无向连通图$G =< V, E >$，其中V代表顶点，E代表顶点之间的边，假设图G中共有n个顶点，m条边。

使用 prim 算法构建最小生成树的时间复杂度可以表示为

$$O(n^2)$$

使用 kruskal 算法构建最小生成树的时间复杂度可以表示为

$$O(mlog_2 m)$$

直观看出，当m较小，即边比较稀疏，我们更应该使用 kruskal 算法，相反对于边较为密集的，我们更应该采用 prim 算法。所以我们可以想如何找到一种自适应算法，让算法能够自适应当前情况选择不同的算法进行处理。

可以将已经创建的最小生成树抽象为一个点，将与局部的最小生成树的点相连的边从总的边集去掉，然后维护剩下的点集和边集。

对于当前的点集和边集我们可以重新衡量当前的图是“稀疏图”还是“稠密图”，即比较n^2和$mlog_2 m$的大小。

当$mlog_2 m \geq n^2$，即为稠密图，则可以使用 prim 算法，反之使用 kruskal 算法。

（2）

没有一种单独的算法是能够完美高效的解决所有问题，要充分了解数据特征，否则我们认为最优的算法也有可能跑到最差的效率。对于当前算法来说，集成的算法集成了两种算法的优点，并且集成的算法可以对于每一种情况找到适合当前数据的较优的算法。

所以，根据效率选择算法才是最为科学的做法。

3. 题目 3

正确性：

首先，手算当$n=1$，$n=2$，$n=3$时的情况。

当$n=1$；程序输出：1；数组$A=[1]$

当$n=2$；程序输出：12，21；数组$A=[2,1]$

当$n=3$；程序输出：123，213，312，132，231，321；数组$A=[1,2,3]$

即，当$n\leq 3$时，符合我们的预期，当n为奇数，执行完程序后数组A不变，当n为偶数，执行完后数组循环右移一位。

利用数学归纳法，因为$n\leq 3$时，算法成立，所以假设当$n\geq 3$

n为偶数时，假设当$n'=n-1$时，算法成立；则可以看出，算法首先输出数组前$n-1$个元素全排列（即$A[1\ldots n-1]$）和数组最后一个元素，并且输出完毕后，前$n-1$个元素位置不变化。之后，因为n为偶数，所以不断交换$A[1\ldots n-1]$和$A[n]$元素并输出它们的全排列和数组最后一个元素。由此，当n为偶数时算法成立。

n为奇数时，假设当$n'=n-1$时，算法成立；则可以看出，算法首先输出数组前$n-1$个元素全排列（即$A[1\ldots n-1]$）和数组最后一个元素，并且输出完毕后，前$n-1$个元素位置循环右移一位。之后，因为n为奇数，所以不断交换$A[1]$和$A[n]$元素并输出它们的全排列和数组最后一个元素。由此，当n为奇数时算法成立。

到此可得，算法 HeapPermute(n)是正确的。

时间效率：

当$n=1$时，$T(n)=1$

当$n\geq 2$时，$T(n)$可以递归地表示成

$$\begin{aligned}T(n)&=n[T(n-1)+2]\\&=n[(n-1)[T(n-2)+2]+2]\\&=n(n-1)(n-2)\ldots+2n+2(n-1)n+2(n-2)(n-1)n\ldots\\&=n!+2(n+(n-1)n+\cdots+n!)\end{aligned}$$

所以，时间复杂度为：$T(n)=O(n!)+O(n^{n-1})$

4. 题目 4

将当前n个实数划分为 2 部分，分别为$1\sim\left\lfloor\frac{n}{2}\right\rfloor$，$\left\lfloor\frac{n}{2}\right\rfloor+1\sim n$

然后分别求出两个区间的最小值进行比价，取较小的作为整体最小值。

伪代码：

```
getMinPos(A, s, e)
// 实现求数组中的最小元素的位置
// 输入：包含n个实数的数组A，数组开始搜索位置s，数组结束搜索位置e，s ≤ e
// 输出：数组A中的最小元素的位置
if s=e then
    //当搜索元素只有 1 个时，直接返回
    return s
end
//mid 表示将数组A划分的中间分隔点
mid ← ⌊(s + e)/2⌋
```

// 分别求取数组A被划分的两部分最小值位置
p1 ← getMinPos(A, s, mid)
p2 ← getMinPos(A, mid+1, e)
// 通过比较，找到最小值的位置
if A[p1]<A[p2] then
    return p1
end
return p2

时间效率：

$$F(n)=\begin{cases}1, & n=1\\ 2F\left(\frac{n}{2}\right)+1, & n>1\end{cases}$$

当$n>1$时，假设$2^k \le n < 2^{k+1}$，则有$\lfloor log_2 n \rfloor = k$

$$\begin{aligned}F(n)&=2F\left(\frac{n}{2}\right)+1\\&=2\left(2F\left(\frac{n}{4}\right)+1\right)+1\\&\cdots\\&=2^{k+1}+(1+2+4+\cdots+2^k)\\&=2^{k+1}+2^{k+1}-1\\&>2n\end{aligned}$$

因为蛮力算法的时间复杂度为$O(n)$，所以减治算法的时间效率不如蛮力算法，并且，减治算法所用空间也大于蛮力算法。

5. 题目 5

约瑟夫斯问题的非递推公式如下：

$$J(n)=1+2n-2^{1+\lfloor log_2 n \rfloor}$$

已知，$m=2$（即隔 1 人杀 1 人）时，有递推式如下：

$$J(2k)=2J(k)-1，n为偶数$$
$$J(2k+1)=2J(k)+1，n为奇数$$

当$m=2$，非递推公式证明如下：

当$n=1$，$J(n)=1+2-2=1$
当$n=2k$（n为偶数），假设当$n=k$时等式成立，有

$$\begin{aligned}&J(2k)=2J(k)-1\\&=2\left(1+2k-2^{1+\lfloor log_2 k \rfloor}\right)-1\\&=1+4k-2^{2+\lfloor log_2 k \rfloor}\\&=1+2*2k-2^{1+\lfloor log_2 2k \rfloor}\\&\text{等式成立}\end{aligned}$$

当$n=2k+1$（n为奇数），假设当$n=k$时等式成立，有

$$J(2k+1) = 2J(k) + 1$$
$$= 2\left(1 + 2k - 2^{1+\lfloor log_2 k \rfloor}\right) + 1$$
$$= 3 + 4k - 2^{2+\lfloor log_2 k \rfloor}$$
$$= 1 + 2 * (2k+1) - 2^{1+\lfloor log_2 2k+1 \rfloor}$$

等式成立

综上所诉，非递推公式$J(n) = 1 + 2n - 2^{1+\lfloor log_2 n \rfloor}$成立。