习题一

1.1 折半查找算法每次选择查找范围的中点执行比较确定递归查找范围，时间复杂度为 O(log$n$)。随机折半查找在查找范围内均匀随机选择一个元素执行比较来确定递归查找范围，用算法语言描述随机折半查找算法并分析算法的时间复杂度。

1.2 理解如下的随机算法，完成后面的问题。

输入：S=$\{s_1,s_2,\ldots,s_n \mid s_i\in\boldsymbol{R}\}$

输出：$min(S, k)$—S 中第 k 小的元素

Random_Select(S,k)

1. 从 S 中随机选择一个元素 s；
2. $S_1=\{s_i \mid s_i\in S, s_i<s\}$, $S_2=\{s_i \mid s_i\in S, s_i>s\}$；
3. IF $|S_1|=k$-1 THEN 返回 s；
4. ELSE IF $|S_1|>k$ THEN 返回 Random_Select(S_1, k);
5. ELSE 返回 Random_Select(S_2, k-$|S_1|$);

（1）该算法属于哪一类随机算法？

（2）证明：存在常数 b<1，使得算法递归过程中所考虑集合的大小的数学期望为 bn。

（3）证明：算法时间复杂度的数学期望为 $O(n)$。

1.3 试设计一个随机算法判定输入的阶分别为 m,n,l 的多项式 $p(x)$,$q(x)$和 $r(x)$是否满足 $p(x)\cdot q(x)=r(x)$。分析随机算法的时间复杂度和获得正确解的概率，判断该随机算法的类别。

1.4. 试设计一个随机算法判定输入的阶分别为 $p\times q$,$q\times r$,$p\times r$ 的矩阵 A,B 和 C 是否满足 $A\cdot B=C$。分析随机算法的时间复杂度和获得正确解的概率，判断该随机算法的类别。

1.5. 证明：最小割问题的如下随机算法输出最小割的概率为$\Omega(1/n^2)$。（ 提示：将该算法与 9.6 节的算法关联起来。）

输入：一个多重无向连通图 G=(V,E)；

输出：G 的一个最小边割。

Random_Mincut

1. 为图 G 的任意边赋予一个随机独立的正权值；
2. 找出 G 的最小生成树 T；
3. 删除 T 中权值最大的一条边得到两棵树 T_1,T_2；
4. 令 T_1 的顶点集为 C，则 T_2 的顶点集为 V-C；
5. $cut=\{uv \mid uv\in E, u\in C, v\in V\text{-}C\}$
6. 输出 cut.

1.6.考虑简单连通图$G = (V;E)$上的最大独立子集问题的如下随机算法。

算法：**IndependentSet()**

输入：$G = (V;E)$

输出：$I\subseteq V$使得$\forall\ uv\in E$均有：$u\in I, v\in I$中至多有一个成立

1. 为V中每个顶点随机分配｛1,2,…,$|V|$｝中唯一标签，不同顶点具有不同标签；

2. $I\rightarrow\varnothing$, $S\leftarrow V$;

---

3. while　$S \neq \varnothing$　do

4.　　$u \leftarrow S$中标签最小的顶点

5．　$I \leftarrow I \cup \{u\}$

6.　　从 S 中删除 u 和 u 的相邻顶点；

7.输出 I

将 IndependentSet 算法输出的集合记为 I。证明：

（1）I 是 $G = (V;E)$的一个独立集；

（2）对$\forall\ u \in V$,　$u \in I$ 的概率等于 $1/d_u$, 其中 d_u 表示 u 在 G 中的度。

（3）计算算法输出的独立集大小的数学期望。

1.7.设 $a_1,a_2,\ldots,a_n$ 是 n 个不同数构成的列表。如果 $i<j$ 且 $a_i>a_j$ 则称 a_i 和 a_j 是倒置的。冒泡排序算法的实质是不断交换列表中相邻的倒置元素，直到列表中没有倒置元素为止。假设冒泡排序算法的输入是一个随机排列，等可能地是 $n!$个排列中的任意一个。确定冒泡排序算法需要交换的倒置元素个数的数学期望。

1.8.有一个函数 $F:\{0,1,\ldots,n\text{-}1\}\rightarrow\{0,1,\ldots,m\text{-}1\}$，且 $F(\ (x+y) \bmod n) = F(x)+F(y) \pmod m$ 对$\forall x,y\in\{0,1,\ldots,n\text{-}1\}$成立。设 $F(x)$存储在一个数组中，数组下标表示自变量的值，数组元素的值表示函数值；由于某种意外，数组中 1/5 的函数值被恶意串改。试设计一个随机算法使其对$\forall z\in\{0,1,\ldots,n\text{-}1\}$算法能够以大于 1/2 的概率计算出正确的 $F(z)$。如果运行算法 3 次，你应该返回什么样的值，此时算法得到正确 $F(z)$的概率有什么变化？