Ex1:PrintPrimes{ //打印 1 万以内的素数

```
print 2， 3；
n ←5；
repeat
if RepeatMillRab(n, ) then print n;
n ←n+2;
until n=10000;
}
```

与确定性算法相比较，并给出 **100~10000** 以内错误的比例。

解：略。

Ex2: 给出与图的独立集问题相关的判定问题的形式化描述，并证明它是 NP 完全的（提示：根据团问题进行规约）

解：

1). 独立集问题相关的判定问题的形式化描述：

独立集定义：G=<V,E>，设 I⊆V，I 中任何点对之间无边，即 E 中每条边至多有 1 个端点在 I 中，则称 V 为独立集。

判定问题：给定图 G 和数 k，判断图 G 中是否存在大小大于等于 k 的独立集。

2). NPC 证明：

可通过规约到最大团问题进行证明，最大团问题是经典的 NPC 问题。

下面使用简单等价规约来证明：

1). 独立集问题 最大团问题

对于图 G，我们可以在多项式时间内构建其补图 G'。假设图 G 存在大小为 k 的独立集，而因为独立集之间没有边连接，所以在补图 G'上该独立集对应的顶点两两之间互相连接，也就是大小为 k 的团。

所以我们可以在多项式时间内将独立集问题转化为寻找补图 G'的最大团问题，只要最大团的大小大于等于 k，则问题成立。

2). 最大团问题 独立集问题

对于图 G，我们可以在多项式时间内构建补图 G'。 假设图 G 的最大团大于等于 k， 因其内部顶点两两相连，所以在补图 G'上，该团表现为大小大于等于 k 的独立集。

所以我们可以在多项式时间内将转化最大团问题为寻找补图 G'的独立集问题，只要独立集的大小大于等于 k，则问题成立。

所以独立集问题可以归约到最大团问题，所以独立集问题是 NPC 问题。

EX3. 证明 **G** 中最大团的 **size** 为α 当且仅当 $\mathbf{G_m}$ 里最大团的 **size** 是 mα 。

解：

1). 充分性：若 G 中最大团的 size 为α，则 G^m 里最大团的 size 是 $m\alpha$。

记 G 的最大团为 C，显然 C^m 是 G^m 的团，因此 G^m 里最大团的 size≥ $m\alpha$。

反正之，如果 size>$m\alpha$，根据鸽巢原理，一定有 $\beta > \alpha$ 个点落在同一个 G 的副本中，这 β 个点显然也是一个团，与 G 的最大团为 C 矛盾。

2). 必要性：若 G^m 里最大团的 size 是 $m\alpha$，则 G 中最大团的 size 为α。

由充分性直接得到。

EX4.完善证明 Th1.9 LPT 算法的近似性能比$R_{LPT} = \frac{4}{3} - \frac{1}{3m}$.

解：

方向一：

假设机器 M_i 上运行的两个任务为 J_{i1} 和 J_{i2}，不妨设 $i1 \le m, i2 > m$。按照轻负载的分配原则，$J_{m+1}, \ldots, J_{2m}$ 将会依次被分配到执行 $J_m, \ldots, J_1$ 的机器上。最终的分配结果满足：对于任意两台机器 M_i 和 M_j，若 $P_{i1} \ge P_{j1}$，则有 $P_{i2} \le P_{j2}$。假设存在一种调度策略 A'，它不同于 LPT，使得 $OPT(I) = A'(I)$。则该策略中，至少存在一对机器 M_i 和 M_j，若 $P_{i1} \ge P_{j1}$，则有 $P_{i2} \ge P_{j2}$。因为:

$$max\{P_{i1} + P_{i2}, P_{j1} + P_{j2}\} \ge max\{P_{i1} + P_{j2}, P_{j1} + P_{i2}\} \tag{4}$$

该不等式右侧即为 LPT 算法的调度策略。则此时的调度算法所用的时间 $OPT(I) = A'(I) \ge A(I)$。由 OPT 的定义可知，$OPT(I) \le A(I)$，所以 $A(I) = OPT(I)$。

方向二：

解： **PPT** 给出了近似比的上界，为了完善证明，我们需要证明这个上界在某些实例下成立。

考虑输入实例 I*，满足如下条件：

$$P_i = 2m - \left\lceil \frac{i}{2} \right\rceil, i = 1,2,\cdots,2m$$

$$P_{2m+1} = m$$

P_1	P_{2m}	P_{2m+1}
P_2	P_{2m-1}	
…		
P_{m-1}	P_{m+2}	
P_m	P_{m+1}	

LPT 运行结果

P_1	P_{2m-2}	
P_2	P_{2m-3}	
…		
P_{m-1}	P_m	
P_{2m-1}	P_{2m}	P_{2m+1}

OPT 运行结果

可见 A(I*)=4m-1,且 OPT(I*)=3m，近似比为$R_{LPT} = \frac{4}{3} - \frac{1}{3m}$。