考试安排

时间：2015-1-19 下午 2:30-4:30

教室：3B115，3B116，3B215

题型：选择+简答（判断）+算法设计

概率算法

Ex.1 若将 y ← uniform(0, 1) 改为 y ← x, 则上述的算法估计的值是什么.

解：如果x 取（0,1）中的随机值，而y=x的话，意味着的比例 k/n代表y=x 这条直线在圆弧内的长度和在正方形内的长度的比例r，所以求的值是4* $1/\sqrt{2}$ ，即 $2\sqrt{2}$ 。

Ex2. 在机器上用 $4\int_0^1\sqrt{1-x^2}\,dx$ **估计 π 值，给出不同的 n 值及精度.**

解：略。

Ex3. 设 a, b, c 和 d 是实数，且 a ≤ b, c ≤ d, f:[a, b] → [c, d]是一个连续函数，写一概率算法计算积分： $\int_a^b f(x)dx$ **.**

解：略。

***EX4. 设 ε , δ 是(0,1)之间的常数，证明：若 I 是** $\int_a^b f(x)dx$ **的正确值，h 是由 HitorMiss 算法返回的值，则当 n ≥ I(1-I)/ ε ^{2} δ 时有：**

Prob[|h-I| < ε] ≥ 1- δ

上述的意义告诉我们：**Prob[|h-I|] ≥ ε] ≤ δ**，即：当 **n ≥ I(1-I)/ ε ^{2} δ** 时，算法的计算结果的绝对误差超过 ε 的概率不超过 δ，因此我们根据给定 ε 和 δ 可以确定算法迭代的次数

$$n=\frac{I(1-I)}{\varepsilon^2\delta}\le\left\lceil\frac{1}{4\varepsilon^2\delta}\right\rceil(\because I(1-I)\le\frac{1}{4})$$

解此问题时可用切比雪夫不等式，将 **I** 看作是数学期望。

解：h 是一个随机变量，记其期望和方差为 E[h]和 Var[h]。根据切比雪夫不等式

$$P(|h - E[h]| \geq \varepsilon) \leq \frac{Var[h]}{\varepsilon^2}$$

显然 E[h]=I。另一方面，在 HitorMiss 算法中，若随机取 n 个点，有 k 个点在积分范围内，则有 h=k / n。因为 n 个点中每一个点，其要么在积分范围内，要么不在积分范围内，因此 k 为二项分布 B(n, I)，其中 I 为点落在积分范围内的概率，Var[k]=n*I(1-I)。又因为 k=h*n，所以

$$Var[h]= I(1-I)/n$$

将 n ≥ I(1-I)/ $\varepsilon^2\delta$ 带入上式即可得证。

EX5. 用上述算法，估计整数子集 1~n 的大小，并分析 n 对估计值的影响.
解：n 越大，估值越准确。

Ex6. 分析 dlogRH 的工作原理，指出该算法相应的 u 和 v.
解： Sherwood 算法的一般过程：
1). 将被解的实例变换到一个随机实例。 //预处理函数 u
2). 用确定算法解此随机实例，得到一个解。
3). 将此解变换为对原实例的解。 //后处理函数 v

dlogRH 是 Sherwood 算法的一个具体应用，dlogRH 为了消除输入实例中 a 的取值对执行时间的影响对其中的 $a=g^x \bmod p$ 作随机预处理，得到与其对应的随机实例 c=u(x, r)，并且对 c 使用确定性算法得到 y ，最后再把随机实例的结果 y 变换为输入实例 a 的解 x=v(y, r)。其中

$$u: u(x, r) = \log_{g,p} c = (r+x) \bmod (p-1)$$
$$v: v(y, r) = (y-r) \bmod (p-1)$$

Ex7. 写一 Sherwood 算法 C，与算法 A, B, D 比较，给出实验结果。
解：略。

Ex8.证明：当放置（k+1)th 皇后时，若有多个位置是开放的,则算法 QueensLV 选中其中任一位置的概率相等。
解：当放置第（k+1）th 皇后时，如果有 n 个位置开放，依次记为$\{S_1,S_2,\ldots,S_n\}$。下面计算选择位置 S_i 的概率 P_i ：S_i 被选中，则 uniform(1,…,i)=1,且对于所有 j>i 有 uniform(1,…,j)=0。显然 uniform(1,…,i)=1 的概率为 1/i, uniform(1,…,j)=0 的概率为(j-1) / j。所以

$$P_i = \left(\frac{1}{i} \times \frac{i}{i+1} \times \ldots \times \frac{n-1}{n}\right) = \frac{1}{n}$$

Ex9. 写一算法，求 n=12~20 时最优的 StepVegas 值。
解：略。

Ex10: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。

近似算法

EX11. 证明 G 中最大团的 size 为α**当且仅当 G^m 里最大团的 size 是**$m\alpha$。

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。

EX12. 完善证明 Th1.9 LPT 算法的近似性能比 $R_{LPT} = \frac{4}{3} - \frac{1}{3m}$**.**

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 $R_{LPT} = \frac{4}{3} - \frac{1}{3m}$。