2015 年

1. 下列陈述错误的是_____。

A 概率算法在同一个输入实例上，每次执行结果不尽相同

B 概率算法在同一个输入实例上，每次执行所花的时间不尽相同

C 有的概率算法对于同一输入实例的不同次运行，可以找到多个不同的正确解

D 概率算法的最坏期望时间是算法执行的上界

2. 下列陈述错误的是_____。

A 数值概率算法一般是求数值计算问题的近似解

B Monte Carlo 总能得到问题的一个解，但该解未必正确

C Las Vegas 算法一定能求出问题的正确解

D Sherwood 算法的主要作用是减少或是消除好的和坏的实例之间的差别

3. Las Vegas 算法的一般形式是

```
Obstinate(x){
    repeat
        LV(x, y, success)
    until success
    return  y
}
```

设 p(x)是 LV 成功的概率，s(x)和 e(x)分别是 LV 成功和失败的期望时间，t(x)是算法 obstinate 找到一个正确解的期望时间，则 t(x)的表达式应该是_____。

A $t(x) = s(x) + e(x)(1-p(x))/p(x)$

B $t(x) = p(x)t(x) + (1-p(x))(e(x) + t(x))$

C $t(x) = p(x)s(x) + (1-p(x))(e(x) + s(x))$

D $t(x) = p(x)s(x) + (1-p(x))(s(x) + t(x))$

4. 若 A 是一个偏真的 MC 算法，则下列陈述正确的是_____。

A 只有 A 返回 true 时解正确

B A 以较大的概率返回 true

C A 返回 true 时解必正确，A 返回 false 时解必错误

D A 返回 true 时解必正确，A 返回 false 时可能产生错误的解

5. 重复调用一个一致的，p-正确的、偏真的 MC 算法 k 次，可以得到一个_____的算法。

A (1-p)-正确

B (1-p)^k-正确

C 1-(1-p)^k-正确

D 正确概率不能确定

6. 一个 MC 算法时一致的、3/5 正确，偏 y_0 的，若要求出错的概率不超过 \epsilon，则重复调用 MC 的次数至少为_____。

A $\lg(1/\epsilon)/\lg(2/5)$

B $\lg(1/\epsilon)/\lg(5/2)$

C $\lg(\epsilon)/\lg(5/3)$

D $\lg(\epsilon)/\lg(3/5)$

7. 在异步环上，一个 O(n^2)的 leader 选举算法按顺时针单向发送消息，假设只有最大的标识符节点可以当选为 leader,则当环上标识符次序为_____时该算法发送的消息数量最多。

A 0,1, … , n-1 随机　　b 逆时针 n-1,n-2,…,0

C 顺时针 0,1, …, n-1　　d 顺时针 n-1,n-2,…,0

8.设正整数 d1,d2,…,dn 是 n 个结点的标识符集合，x = min(d1,d2,…,dn),y = max(d1,d2,…,dn),则同步环上非均匀的 leader 选举算法的时间复杂性是_____

A O(n)　b O(xn)　c (yn)　d O(nlogn)

9.在下述因素中，已知有 3 个阻碍分布式系统了解系统全局状态，与全局状态无关的是___

A 非及时的通信 b 相对性影响 c 中断　d 算法的正确性

10. 下述说法错误的是___

A 异步系统中的消息延迟是不确定的

B 分布式算法的消息复杂性是指在所有合法的执行上发送消息总数的最大值

C 在一个异步算法中，如果不存在错误，则算法的执行只取决于初始配置

D 分布式系统终止是指系统中所有结点处于终止状态，且没有消息在传输

二．简要回答下述问题（55 分）

1 构造一个 16 节点的环，使其高度对称，并给出所有序等价的连续片段。

0　0000　0000　0
1　0001　1000　8
2　0010　0100　4
3　0011　1100　12
4　0100　0010　2
5　0101　1010　10
6　0110　0110　6
7　0111　1110　14
8　1000　0001　1
9　1001　1001　9
10　1010　0101　5
11　1011　1101　13
12　1100　0011　3
13　1101　1011　11
14　1110　0111　7
15　1111　1111　15

长度为 1 的有序等价的连续片段:

(0), (8), (4), (12), (2), (10), (6), (14), (1), (9), (5), (13), (3), (11), (7), (15)

长度为 2 的有序等价的连续片段:

(0, 8),(4, 12),(2, 10),(6, 14),(1, 9), (5, 13), (3, 11), (7, 15);

(8, 4), (12, 2), (10, 6), (14, 1), (9, 5), (13, 3), (11, 0)

长度为 4 的有序等价的连续片段:

0, 8, 4, 12 || 2, 10, 6, 14 || 1, 9, 5, 13 || 3, 11, 7, 15;
15 0, 8, 4 || 12, 2, 10, 6 || 14, 1, 9, 5 || 13, 3, 11, 7;
7, 15 0, 8, || 4, 12, 2, 10, || 6, 14 1, 9, || 5, 13 3, 11;
11, 7, 15 0, || 8, 4, 12, 2, || 10, 6, 14, 1, || 9, 5, 13 3;
长度为 8 的有序等价的连续片段:
0, 8, 4, 12 2, 10, 6, 14 || 1, 9, 5, 13 || 3, 11, 7, 15;
15 0, 8, 4 12, 2, 10, 6 || 14, 1, 9, 5 , 13, 3, 11, 7;
7, 15 0, 8, 4, 12, 2, 10, || 6, 14 1, 9, 5, 13 3, 11;
11, 7, 15 0, 8, 4, 12, 2, || 10, 6, 14, 1, 9, 5, 13, 3;
3, 11, 7, 15 0, 8, 4, 12, || 2 10, 6, 14, 1, 9, 5, 13 ;
13, 3 11, 7, 15 0, 8, 4, || 12, 2 10, 6, 14, 1, 9, 5;
5, 13, 3 11, 7, 15 0, 8, || 4 12, 2 10, 6, 14, 1, 9 ;
9, 5, 13, 3 11, 7, 15 0, || 8, 4 12, 2 10, 6, 14, 1;
长度为 16 的有序等价的连续片段(0, 8, 4, 12, 2, 10, 6, 14, 1, 9, 5, 13, 3, 11, 7, 15)

2 已知事件 e1,e2,e3 和 e4 的向量时戳分别为（2,3,0,0），（1,2,0,0），（0,0,1,1），（3,6,4,2），请找出所有因果关系的事件对。

（1,2,0,0）<v （2,3,0,0），e2 在因果序上先于 e1
（2,3,0,0）<v （3,6,4,2), e1 在因果序上先于 e4
（1,2,0,0）<v （3,6,4,2), e2 在因果序上先于 e4
（0,0,1,1）<v （3,6,4,2） e3 在因果序上先于 e4

3 若将消息复杂度为 O(nlgn)的异步环选举算法（在阶段 1 向节点的 2 邻居发送 Prob 消息）修改为只向其中一个方向发送 Prob 消息，请问修改后算法的消息复杂度是多少？如何对其做进一步的修改使得消息复杂度仍然为 O(nlgn)。

The algorithm first computes the smallest identity and makes it known to each process, then the process with that identity becomes leader and all others are defeated. The algorithm is more easily understood if one first considers it as if it were an algorithm executed by the *identities* rather than by the processes. Initially each identity is *active*, but in each round some identities become *passive* as will be shown later. In a round an *active* identity compares itself with the two neighboring *active* identities in clockwise and anticlockwise directions; if it is a local minimum, it survives the round, otherwise it becomes *passive*. Because all identities are different, an identity next to a local minimum is not a local minimum, which implies that at least half of the identities do not survive the round. Consequently, after at most $\log N$ rounds only one *active* identity remains, which is the winner.

This principle can be elaborated in a straightforward manner in bidirectional networks, as is done in Franklin's algorithm [Fra82]. In directed rings messages can be sent only clockwise, which makes it difficult to obtain the next active identity in the clockwise direction; see Figure 7.6. Identity q must be compared with r and p; identity r can be sent to q, but identity p would have to travel against the direction of the channels in order to arrive at q. To make a comparison with both r and p possible, identity q is sent (in the direction of the ring) to the process holding identity p and r is forwarded not only to the process holding q but also, further, to the process holding p. If q is the only *active* identity at the beginning of a round, the first identity that q encounters on its tour equals q (i.e., $p = q$ in this case). When this situation occurs this identity is the winner of the election.

Claim 7.9 *If a round starts with exactly one active process p, with current identity ci_p, the algorithm terminates after that round with $win_q = ci_p$ for each q.*

Proof. The $\langle$ **one**, ci_p $\rangle$ message of p is relayed by all processes and finally received by p. Process p obtains $acn_p = ci_p$ and sends a $\langle$ **smal**, acn_p $\rangle$ message around the ring, which causes each process q to exit the main loop with $win_q = acn_p$. □

The algorithm terminates in each process and and all processes agree on

---

the identity of the leader (in the variable win_q); this process is in the state *leader* and all other processes are in the state *lost*.

There are at most $\lfloor \log N \rfloor + 1$ rounds, in each of which exactly $2N$ messages are exchanged, which proves that the message complexity is bounded by $2N \log N + O(N)$. This proves Theorem 7.7. □

4. 对于一个优化问题Π，最佳可达性能比为$R_{min}(\Pi)$（定义如下）分别为何值时，问题Π易于近似和难于近似?

$$R_{min}(\Pi) = \inf\{(r \geq 1 | \exists)\text{的多项式时间算法 A 使得} R_A^{\infty} \leq r\}$$

当$R_{min}(\Pi)$的值为 1 时，问题Π是容易近似的。当$R_{min}(\Pi)$的值为无界$(+\infty)$时，问题Π是难于近似的。

5. 装箱问题是将 n 件物品放入尽可能少的若干个容量为 1 的箱子中。不妨设实例 I 中，物品$item_j(1 \leq j \leq n, n = 6)$的大小依次为：0.4，0.3，0.6，0.7，0.8，0.2，请分别给出实例 I 的最优解和采用首次适应（first fit）策略得到的近似解的值 OPT(I)和 A(I)，并给出解的构造，以及近似比$R_{FF}(I)$。

近似解的值$OPT(A) = 3$ 解的构造为$\sigma = \{\{item_1, item_3\}, \{item_2, item_4\}, \{item_5, item_6\}\}$

采用首次适应 FF 策略得到的近似解$A(I) = 4$,
解的构造为: $\{\{item_1, item_2, item_6\}, \{item_3\}, \{item_4\}, \{item_5\}\}$

近似比为: $R_{FF} = \frac{A(I)}{OPT(A)} = \frac{4}{3}$

5. 说明为什么用 MST 启发解△TSP 时，其近似比是 2。
设$d(H)$:表示 H 里所有边的长度之和。
给定图 G(V, E)和距离函数 d，以及 G 上的最小生成树 T。

图 G 最优解构造是一个哈密顿圈，而任何哈密顿圈是由 1 棵生成树和 1 条多余的边构成的，所以$OPT(G) \geq d(\text{某生成树}) \geq d(T)(1)$

根据 MST 启发，会复制 T 上的所有边一次，构成新的多重图 T'，然后得出欧拉圈 ET，所以： $d(ET) = d(T') = 2d(T)(2)$

近似算法最后得到哈密顿圈是由 EF 短路而来，所以$A_{MST}(G) \leq d(ET)$ (3)

根据(1)(2)(3)式可以得到$A_{MST}(G) \leq 2OPT(G), R_{MST}^{\infty} \leq \frac{A_{MST}(G)}{OPT(G)} = 2$

三 算法题（25 分）

1. 设一个同步匿名的单向环有 n 个结点，每个结点均知道 n，每个节点的初始均状态相同，每个结点上的程序相同且开始于同一时刻。
   （1） 请问是否存在一个确定的算法选出一个 leader？简述理由。
   （2） 试设计一个概率的 leader 选举算法。
   （3） 请问你设计的概率算法属于哪一类算法？

(1) 对于同步环上的 leader 选举， 不存在非均匀的匿名算法。

1. *初始状态相同：在一个匿名环中，处理器间始终保持对称，若无某种初始的非对称(如，标识符唯一)， 则不可能打破对称。 在匿名环算法里， 所有处理器开始于相同状态。*
2. *在同步系统中， 一个算法以轮的形式进行，根据引理 3.1 在环 R 上算法 A 的容许执行里， 对于每一轮 k，所有处理器的状态在第 k 轮结束时是相同的。*

由反证法，假设若在某轮结束时， 一个处理器宣布自己是 leader(进入选中状态)， 则其它处理器亦同样如此， 这和 A 是一个 leader 选举算法的假定矛盾!

（2）算法由若干个 phase 构成，每个 phase 包括 n 轮，可用 phase 和轮控制算法流程。每个结点可以设置一个随机数发生器 uniform(1...m),这里 m 是局部变量，初值等于 n。

每个结点上的随机数发生器 uniform(1...m)产生随机数x_i*当作自己的 id*

在第 i 阶段开始

1. *若一个结点的 id 是 i，则该结点绕环发送一个包含信息 i 的 msg;*
2. *若一结点的 id 不是 i， 且它收到一个 msg， 则它转发此 msg 后变为 non-leader，变成 non-leader 之后能转发 msg。*
3. *若一个结点的 id 是 i， 同时收到一个 msg，则本地变量 count 记录收到了多少个 msg。*

id 为 i 的结点收到了 msg 数量,代表当前系统中产生了最小 id 的结点个数。此时 id 为 i 的结点把 Phase 变为 1，m 变成 count。重新执行以上过程，直到 id 为 i 的结点只收到一个 msg,此时该结点为 leader,同时绕环发送终止 msg,收到终止 msg 的 non-leader 终止运行。

（3）Las Vegas 算法。因为自己提出的算法一定能获得正确的解，但是随机算法可能会以极低的概率一直产生相同x_i*，导致不能产生最终解。*