

算法历年大题汇总

一. 简答题

1. 在分布式算法中, bit 复杂性是在算法发送的所有消息中 bit 的总数; 消息链复杂性是在算法的任何执行中最长消息链的长度, 若某消息链是 m_1, m_2, \dots, m_k , 若 m_i 在因果关系上领先于 m_{i+1} , 则该消息链长度为 k 。请问这两种复杂性分别属于消息复杂性和时间复杂性中的哪一种? 并简述理由。

答:

bit 复杂性属于通信复杂性, 消息链复杂性属于时间复杂性; 若在一个分布式算法中每个 msg 信息的 bit 数目相同, 则 msg 的个数就等于 bit 的总数除以一个 msg 的 bit 数目, 则 bit 复杂性可以等价为 msg 复杂性; 消息链复杂性是最长消息链的长度, 在同步系统中它就是最大轮数, 异步系统中假定任何执行的 msg 延迟至多是一个单位时间, 它就是计算直到终止时间的最大运行时间, 在同, 异步系统中皆为时间复杂性。

2. 已知事件 e_1, e_2, e_3, e_4 的向量时间戳分别是 $(1, 0, 0, 0)$ 、 $(3, 5, 0, 0)$ 、 $(0, 0, 1, 2)$ 、 $(3, 6, 4, 3)$, 与 e_3 有因果关系的是哪个事件? 若该事件发生在 e_3 之前, 会怎样?

答:

e_4 , 处理器会抑制过早到达的 e_4 msg (不发送 e_4), 直到 e_3 msg 达到, 才会将 e_3, e_4 一起发送。

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

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

答:

该优化问题渐进性能比上界集合中的下确界, 当 R_{min} 为:

(1) 1 时, 因为渐进性能比大于等于 1, 所以渐进性能比可以无限接近 1, 则易于近似;

(2) 正无穷大时, 因为渐进性能比大于等于 1, 所以渐进性能比也为无穷大, 则难于近似;

4. 对于某优化问题, 什么情况下其近似算法的绝对性能比和渐进性能比相同?

答:

具有 Scaling 性质的问题, 近似算法的绝对性能比和渐进性能比是相同的。

5.装箱问题是将 n 件物品放入尽可能少的若干个箱子中。不妨设每个箱子的容量为 1, 物品 $I_j (1 \leq j \leq n), n = 6$ 的大小依次为: 0.5、0.6、0.3、0.7、0.5、

0.4, 请给出其最优解, 以及采用首次适应策略 (First Fit) 得到的近似解。这里, 解是指使用了几个箱子, 每个箱子放了哪些物品。

答:

FF 的近似解: 4 个箱子; $x_1=0.5+0.3; x_2=0.6+0.4; x_3=0.7; x_4=0.5$

最优解: 3 个箱子; $x_1=0.3+0.7; x_2=0.4+0.6; x_3=0.5+0.5$

6.若要讲一个偏 y 的, 55%-正确的一致 MC 算法改进到 95-正确的算法, 需要重复调用 MC 算法多少次?

答:

$1-(1-55\%)^n \geq 95\%$, $n=3$ 时: 左式=0.91; $n=4$ 时, 左式=0.959

7.在分布式算法的时间复杂性和 one-time 复杂性中, 一个 msg 的延迟分别假定至多为 1 个时间单位和恰好一个时间单位, 但有时后者是前者的一个下界, 为什么? 请举例说明

答:

考虑运行在环上的分布式算法的 1-time 时间复杂性和时间复杂性。

<1> 1-time 时间复杂性:

满足条件 O2: 发送和接收一个 msg 之间的时间恰好是一个时间单位, 每个阶段节点转发消息都是同步进行, 从而 1-time 时间复杂度仅与环直径相关, 为 $O(D)$ 。

<2> 时间复杂度:

满足条件 T2: 一个 msg 的发送和接收之间的时间至多为一个时间单位, 即为 $O(1)$ 。节点转发消息并非同步进行, 消息转发轨迹可能呈链状结构, 时间复杂性与环节点数相关, 为 $O(n)$ 。

例如: echo 协议, 即应答协议, 主要用于调试和检测中, 是路由也是网络中最常用的数

据包, 可以通过发送 echo 包知道当前的连接节点有哪些些路径, 并且通过往返时间能得

出路径长度。echo 算法的实现, 如果转发消息同步进行, 则对应 1-time 时间复杂性, 为

$O(D)$; 如果不同步转发消息, 网络路径可能呈链状结构, 即对应时间复杂度 $O(N)$ 。

Note: 考虑时间复杂度, 任一节点可以在 $O(d)$ 时间内将询问包发送到网络上的其它节点, 但却可能需要 $O(N)$ 的时间接收其它节点发来的响应包。

8.对于同步环, 在一个均匀的 leader 选举算法中, 为什么一个 id 为 i 的 msg 是以 2^i 速率被转发的? 其目的是什么?

答:

同步环的 leader 选举算法是选最小的 id, 不同的 id 的 msg 以 2^i 速率转发时, leader 的 msg 的转发速率(延时)最小, 则可以使得其他非 leader 的转发 msg 被淹没, 降低消息复杂度。

9. 设 $F(x)$ 是一个 MC 算法，若 $F(x)$ 以大于 $1/2$ 的概率返回 true, 且放回 true 时算法正确，则下述算法 $F2(x)$ 是偏真的还是偏假的？请分析 $F2(x)$ 的出错概率至多是多少？

```

F2(x)
{
    If F(x) then
        return true;
    else return F(x);
}

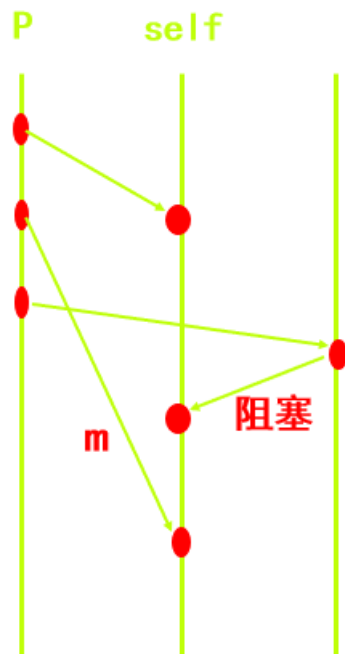
```

答：

偏真；第一次 $F(x)$ 返回 false，第二次也放回 false 则 $1/2 * 1/2 = 1/4$

10. 试举例说明 Casual Msg delivery 算法可能出现的死锁情况，并分析为什么该算法通常被应用于组播通信的一部分？

答：若一节点长时间不发送你要的 msg，则会发生死锁。
本地节点的初始化在本地节点完成，有处理阻塞的机制。



二. 算法题

1、设网络的生成树已经建立，各个节点 P_i 的 id 为 i ，并持有初值 x_i ，且 id 和持有的初值均互不相同，试写一个分布式算法使得根节点知道树中持有初值最大的节点，以及持有初值最小的节点。

答：生成树上的 leader(初值最大，最小)选举算法

节点 i 拥有局部变量 $max, min, max_x, min_x, lx, id(i)$ ，拥有所有子节点的集合 $child(i)$ ，拥有父节点 $parent(i)$

设有 n 个节点，root 节点 $id=0$ ，则算法如下：

Code for P_i , $0 \leq i \leq n-1$

var $child(i)[], parent(i), id(i), max=id(i), min=id(i), lx, max_x=lx, min_x=lx$;

if ($i \neq 0$)

```

{
    upon receiving<max,min,max_x,min_x> from child Pj
        将受到的值与本地值比较,重新赋值 max,min,max_x,min_x;
        child(i)[]=child(i)[]-Pj; //将 Pj 从 child 中删去
        if child(i)[]为空
            send<max,min,max_x,min_x> to parent(i);
}
if(i==0)
{
    upon receiving<max,min,max_x,min_x> from child Pj
        将受到的值与本地值比较,重新赋值 max,min,max_x,min_x;
        child(i)[]=child(i)[]-Pj; //将 Pj 从 child 中删去
}
if(i 为叶子节点)
{
    send<max,min,max_x,min_x> to parent(i);
}

```

2. 设集合 S 和 T 中各有 n 个互不相同的元素，要求：

(1) 写一 Monte Carlo 算法判定 S 和 T 是否相等

(2) 分析算法出错的概率

(3) 算法是否有偏，若有偏，偏什么？

答：

(1)

STequal(S,T)

```

{
    a=uniform(S);
    for i from 1 to n
    {
        if a=T[i];
            return true;
    }
    return false;
}

```

(2) 设有 x 个元素相同

x/n

(3) 偏假

3. 设一个同步匿名的单向环有 n 个节点，每个节点均知道 n，每个节点初始状态相同，每个节点上的程序开始于同一时刻。

(1) 请问是否存在一个确定的算法选出一个 leader，请简述理由

(2) 试设计一个概率的 leader 选举算法。提示：该算法由若干个 phase 构成，

每个 phase 包括 n 轮，可用 phase 和轮控制算法流程。每个节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 。

(3) 请问你设计的概率算法属于哪一类算法？

答：

(1) 由 Lemma3.1 可得。(同步匿名非均匀)

假设 R 是大小为 $n > 1$ 的环（非均匀）， A 是其上的一个匿名算法，它选中某处理器为 leader。因为环是同步的且只有一种初始配置，故在 R 上 A 只有唯一的合法执行。

Lemma3.1: 在环 R 上算法 A 的容许执行里，对于每轮 k ，所有处理器的状态在第 k 轮结束时是相同的。**Note:** 每个处理器同时宣布自己是 Leader！

(2)

n 个节点每个节点随机产生一个 $(1..n)$ 的随机数作为自己的 id，然后将它发送，若绕场一周后回到了该节点，该节点就用那个 id；

否则重新产生随机数，直到绕场一周回到该点；

对于每个节点若收到的 msg 中的 id 和自己产生的 id 一样就没收它，反之转发该 msg；

每个节点重复上述过程直至 n 节点的 id 均不相同；

于是得到一个同步非匿名环，可选举一个 id 最大的 leader。

(3)

4. 量子运动的随机聚合过程可用量子读博来描述。其规则是：

a) 开始时， A 和 B 的赌本分别是 x 和 y

b) 每次通过一枚神奇的硬币来决定输赢，设正面 A 赢，反面 B 赢，但每次扔出硬币的正反面概率正比于 A 和 B 当前的赌本

c) 每次的输家按固定的比例 k 从自己的赌本中付给赢家

d) 设最小的赌本单位为 1，若输家当前的赌本小于等于 1，他付出自己的赌本后，游戏结束。

要求：

(1) 写一算法实现赌博游戏

(2) A 和 B 最终的输赢取决于什么？

(3) 请分析 A 、 B 最终输赢的概率

答：

```
while(x>1 and y>1)
{
    a=uniform(1,x+y);
    If(a<=x)
    {
        x=x+0.1y;
        y=0.9y;
    }
    else
    {
        y=y+0.1x;
        x=0.9x;
    }
}
//end while
If(x<=1)
{
    y=y+x;
}
If(y<=1)
{
    x=y+x;
}
```