中国科学技术大学
University of Science and Technology of China

2021 春算法理论 复习与习题课

2021-07-09

考试安排

- **考试形式**

  闭卷考试，不允许任何电子设备和书籍、资料带入考场

- **范围**

  概率算法（50-60%）、分布式算法（40-50%）

- **考试时间**

  7月20日9：30-11：30

- **题型分布**

  选择题 （10 *3）+ 计算与简答题（4*10）+ 算法设计题（2*15）

复习要点

概率算法

概率算法期望时间和平均时间的区别

– 确定算法的平均执行时间

  输入规模一定的所有输入实例是等概率出现时，算法的平均执行时间。

– 概率算法的期望执行时间

  反复解同一个输入实例所花的平均执行时间。

因此，对概率算法可以讨论如下两种期望时间

① 平均的期望时间：所有输入实例上平均的期望执行时间

② 最坏的期望时间：最坏的输入实例上的期望执行时间

① 快速排序中的随机划分

要求学生写一算法，由老师给出输入实例，按运行时间打分，大部分学生均不敢用简单的划分，运行时间在1500-2600ms，三个学生用概率的方法划分，运行时间平均为300ms。

② 8皇后问题

系统的方法放置皇后(回溯法)较合适，找出所有92个解 $O(2^n)$，若只找92个其中的任何一个解可在线性时间内完成O(n)。

随机法：随机地放置若干皇后能够改进回溯法，特别是当n较大时，可提升效率

时间复杂度：

排列树：$O(p(n) * n!)$ —— 排列

子集树：$O(p(n) * 2^n)$ ——二叉树

可复选的排列树：$O(p(n) * n^n)$

③ 判断大整数是否为素数

确定算法无法在可行的时间内判断一个数百位十进制数是否素数，否则密码就不安全。

概率算法将有所作为：若能接受一个任意小的错误的概率

MC算法计算定积分的值

Monte Carlo积分(但不是指我们定义的MC算法)

1、概率算法1

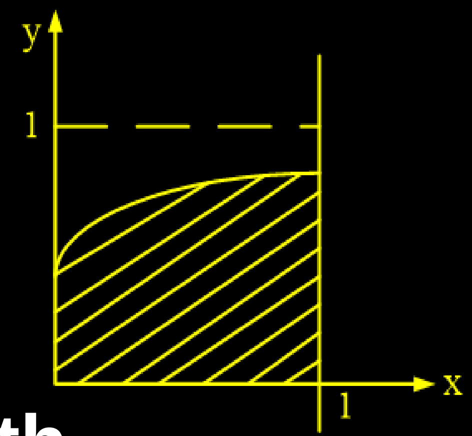
设f: [0, 1] → [0, 1]是一个连续函数，则由曲线y=f(x), x轴, y轴和直线x=1围成的面积由下述积分给出：

$$S = \int_0^1 f(x)dx$$

向单位面积的正方形内投镖n次，落入阴影部分的镖的数目为k，则

$$\frac{k}{n} = \frac{S}{1} => S = k/n$$

显然，只要n足够大 $S -> k/n$



用大量的随机试验模拟来估计/逼近真实的Ground truth

21

Sherwood算法的随机化预处理

将选择和排序的确定算法修改为Sherwood算法很简单，但是当算法较复杂，例如它是一个缺乏文档资料的软件包的一部分时，就很难对其进行修改。注意，只有当该算法平均时间性能较优，但最坏性能较差时，才有修改的价值。

一般方法是：

① 将被解的实例变换到一个随机实例。// 预处理

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

LV算法和Sherwood算法比较

Ch.4 Las Vegas 算法

- Las Vegas和Sherwood算法比较

Sherwood算法一般并不比相应的确定算法的平均性能优

Las Vegas一般能获得更有效率的算法，有时甚至是对每个实例皆如此

Sherwood算法可以计算出一个给定实例的执行时间上界

Las Vegas算法的时间上界可能不存在，即使对每个较小实例的期望时间，以及对特别耗时的实例的概率较小可忽略不计时。

- Las Vegas 特点

可能不时地要冒着找不到解的风险，算法要么返回正确的解，要么随机决策导致一个僵局。

若算法陷入僵局，则使用同一实例运行同一算法，<u>有独立的机会求出解</u>。

成功的概率随着执行时间的增加而增加。

LV算法

- 算法的一般形式

LV(x, y, success) —— x是输入的实例，y是返回的参数，success是布尔值，true表示成功，false表示失败

p(x) —— 对于实例x，算法成功的概率

s(x) —— 算法成功时的期望时间

e(x) —— 算法失败时的期望时间

一个正确的算法，要求对每个实例x，p(x)>0，更好的情况是:

$$\exists 常数\ \delta > 0, p(x) \geq \delta$$

LV算法

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

设t(x)是算法obstinate找到一个正确解的期望时间，则

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

LV成功的概率　LV失败的概率

$$t(x)=s(x)+\frac{1-p(x)}{p(x)}e(x)$$

若要最小化t(x)，则需在p(x)，s(x)和e(x)之间进行某种折衷，例如，若要减少失败的时间，则可降低成功的概率p(x)。

八皇后问题

- 问题及改进
  - 消极：LV算法过于消极，一旦失败，从头再来
  - 乐观：回溯法过于乐观，一旦放置某个皇后失败，就进行系统回退一步的策略，而这一步往往不一定有效。
  - 折中：会更好吗？一般情况下为此。

先用LV方法随机地放置前若干个结点，例如k个。

然后使用回溯法放置后若干个结点，但不考虑重放前k个结点。

若前面的随机选择位置不好，可能使得后面的位置不成功，如若前两个皇后的位置是1、3。

随机放置的皇后越多，则后续回溯阶段的平均时间就越少，失败的概率也就越大。

纯回溯时间： 40ms

stepVegas=2 or 3： 10ms （平均）

纯贪心LV： 23ms （平均）

结论： 一半略少的皇后随机放置较好。

整数的因数分解

设n是一个大于1的整数，因数分解问题是找到n的一个唯一分解：$n = p_1^{m_1} p_2^{m_2} \cdots p_k^{m_k}$

这里 m_i 是正整数，且 $p_1 < p_2 < \cdots < p_k$ 均为素数。

若n是合数，则至少有1个非平凡的因数(不是1和n本身).

设n是一个合数，n的因数分解问题，即找n的非平凡因数，它由两部分构成：

① prime(n) ——判定n是否为素数，可由Monte Carlo算法确定。

② split(n)——当n为合数时，找n的一个非平凡的因数。

整数的因数分解（k-平滑定义）

4. 如何确定a和b使 $a^2 \equiv b^2 (\bmod n)$，来对n因数分解。

- Def. k-平滑：

　　若一个整数x的所有素因子均在前k个素数之中，则x称为k-平滑的。

- 例如：$120 = 2^3 \times 3 \times 5$是3-平滑的

　　35=5×7不是3-平滑的，∵7是第四个素数

　　∴它是4-平滑的，也是5-平滑的…

当k较小时，k-平滑的整数可用朴素的split算法进行有效的因数分解。Dixon算法可以分为3步确定a和b。

Monte Carlo算法

某些MC算法的参数不仅包括被解的实例，还包括错误概率的上界。因此，这样算法的时间被表示为实例大小及相关可接受的错误概率的函数。

- 基本思想：为了增加一个一致的、p-正确算法成功的概率，只需多次调用同一算法，然后选择出现次数最多的解。

例：设MC(x)是一个一致、75%-correct的MC算法，考虑下述算法：

```
MC3(x){
    t←MC(x); u←MC(x); v←MC(x);
    if  t=u or t=v then return t;
    else return v;
}
```

Monte Carlo算法

该算法是一致的和27/32-correct的(约84%)

pf: 相容性（一致性）易证。

∵ t、u、v正确的概率为75%=3/4=p

∴ 错误的概率为q=1/4.

1)若t、u、v均正确， ∵MC是一致的 ∴ t=u=v，则MC3 返回的t正确，此概率为：$(3/4)^3$

2)若t、u、v恰有两个正确则MC3返回 $\begin{cases} t\text{正确} & \text{if } t=u \text{ or } t=v \\ v\text{正确} & \text{if } u=v \end{cases}$

此概率为：$C_3^2 p^2 q^1 = 3\times(3/4)^2(1/4)$

3)若t、u、v恰有一个正确，则只有v正确时，MC3返回正确答案，此概率为：$pq^2 = (3/4)(1/4)^2$

严格的说，当v正确，只有两个错误的解t和u不相等时，才有可能成功，因此MC3成功的概率为：

$$\left(\frac{3}{4}\right)^3 + 3\left(\frac{3}{4}\right)^2\left(\frac{1}{4}\right) + \frac{3}{4}\left(\frac{1}{4}\right)^2 = \frac{27}{32} + \frac{3}{64} > \frac{27}{32} \approx 84\%$$

多运行2次（共3次）使成功率 75%↗84%

复习要点

分布式算法

分布式系统的概念

§ 1.1 分布式系统

- **Def：**一个分布式系统是一个能彼此通信的单个计算装置的集合（计算单元：硬——处理器；软——进程）

  包括：紧耦合系统----如共享内存多处理机

  松散系统-----cow、Internet

- **与并行处理的分别**(具有更高程度的不确定性和行为的独立性)
  - 并行处理的目标是使用所有处理器来执行一个大任务
  - 而分布式系统中，每个处理器一般都有自己独立的任务，但由于各种原因（为共享资源，可用性和容错等），处理机之间需要协调彼此的动作。

- **分布式系统无处不在，其作用是：**

  ①共享资源

  ②改善性能：并行地解决问题

  ③改善可用性：提高可靠性，以防某些成分发生故障

分布式系统的概念

- **分布式系统面临的困难**
  - **异质性：**软硬件环境
  - **异步性：**事件发生的绝对、甚至相对时间不可能总是精确地知道
  - **局部性：**每个计算实体只有全局情况的一个局部视图
  - **故障：**各计算实体会独立地出故障，影响其他计算实体的工作。

分布式系统的概念

§2.1.1 系统

- **容许的执行：**指无限的执行。
  因为轮的结构，所以
  每个处理器执行无限数目的计算步，
  每个被发送的msg最终被传递
- **同步与异步系统的区别**
  在一个无错的同步系统中，一个算法的执行只取决于初始配置
  但在一个异步系统中，对于相同的初始配置及无错假定，因为处理器步骤间隔及消息延迟均不确定，故同一算法可能有不同的执行。

分布式系统生成树构造

§2.3 构造生成树

- **msg复杂性**
  因为每个结点在任一信道上发送M不会多于1次，所以每个信道上M至多被发送两次(使用该信道的每个处理器各1次)。
  在最坏情况下：M除第1次接收的那些信道外，所有其他信道上M被传送2次。因此，有可能要发送2m-(n-1)个msgs。这里m是系统中信道总数，它可能多达n(n-1)/2。
- **时间复杂性：**O(D)　D—网直径

2.构造生成树

对于flooding稍事修改即可得到求生成树的方法。

分布式系统生成树构造

§2.3 构造生成树

- **为何无环？(无环)**

  假设有一环，$p_{i1},\ldots p_{ik}p_{i1}$，若$p_i$是$p_j$的孩子，则$p_i$在$p_j$第1次收到M之后第1次收到M。因每个处理器在该环上是下一处理器的双亲，这就意味着p_{i1}在p_{i1}第1次接收M之前第1次接收M。矛盾！

- **复杂性**

  显然，此方法与淹没算法相比，增加了msg复杂性，但只是一个常数因子。在异步通信模型里，易看到在时刻t，消息M到达所有与p_r距离小于等于t的结点。因此有：

Th2.7 对于具有m条边和直径D的网络，给定一特殊结点，存在一个msg复杂性为O(m)，时间复杂性为O(D)的异步算法找到该网络的一棵生成树。

环选举-异步环选举

§ 3.3 异步环

- 在非匿名算法中，均匀（一致性）和非均匀（非一致性）的概念稍有不同
  - ① **均匀算法：**每个标识符id，均有一个唯一的状态机，但与环大小n无关。而在匿名算法中，均匀则指所有处理器只有同一个状态。（不管环的规模如何，只要处理器分配了对应其标识符的唯一状态机，算法就是正确的。）
  - ② **非均匀算法：**每个n和每个id均对应一个状态机，而在匿名非均匀算法中，每个n值对应一个状态机。（对每一个n和给定规模n的任意一个环，当算法中每个处理器具有对应其标识符的环规模的状态机时，算法是正确的。）

  下面将讨论msg复杂性：$O(n^2) \to O(n\log n) \to \Omega(n\log n)$

下界

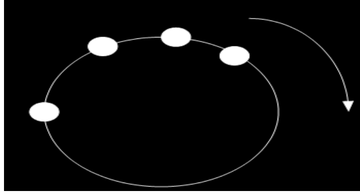
§ 3.3.1 一个$O(n^2)$算法

Le Lann、Chang和Roberts给出，LCR算法

- **基本思想**
  - ① 每个处理器P_i发送一个msg(自己的标识符)到左邻居，然后等其右邻居的msg
  - ② 当它接收一个msg时，检验收到的id_j，若$id_j>id_i$，则P_i转发id_j给左邻，否则没收id_j(不转发)。

  - ③ 若某处理器收到一个含有自己标识符的msg，则它宣布自己是leader，并发送一个终止msg给左邻，然后终止。
  - ④ 当一处理器收到一个终止msg时，向左邻转发此消息，然后作为non-leader终止。

  因为算法不依赖于n，故它是均匀的。



i—表示id 单向

环选举-异步环选举

§3.3.2 一个O($n$lg$n$)算法

■ **基本思想**

算法按阶段执行，在第l阶段一个处理器试图成为其2^l-邻接的临时leader。只有那些在l-th阶段成为临时领袖的处理器才能继续进行到$(l+1)$th阶段。因此，l越大，剩下的处理器越少。直至最后一个阶段，整个环上只有一个处理器被选为leader。

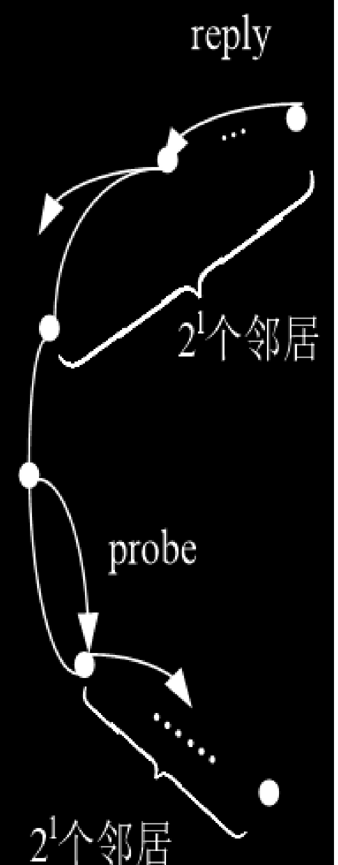
■ **具体实现**

① **phase0:** 每个结点发送1个probe消息(其中包括自己的id)给两个1-邻居，若接收此msg的邻居的id大于消息中的id，则没收此msg；否则接收者发回一个reply msg。若一个结点从它的两个邻居收到回答msg reply，则该结点成为phase0里它的1-邻居的临时leader，此结点可继续进行phase1。

环选举-异步环选举

② **phase l**：在l-1阶段中成为临时leader的处理器P_i发送带有自己id的probe消息至它的2^l邻居。若此msg中的id小于左右两个方向上的$2*2^l$个处理器中任一处理器的id，则此msg被没收。若probe消息到达最后一个邻居而未被没收，则最后一个处理器发送reply消息给P_i，若P_i从两个方向均接收到reply消息，则它称为该阶段中2^l邻居的临时leader，继续进入下一阶段。

③ **终止**：接收到自己的probe消息的结点终止算法而成为leader，并发送一个终止msg到环上。



环选举-异步环选举

④ **控制probe msg的转发和应答**

probe消息中有三个域：<prob, id, *l*, hop>

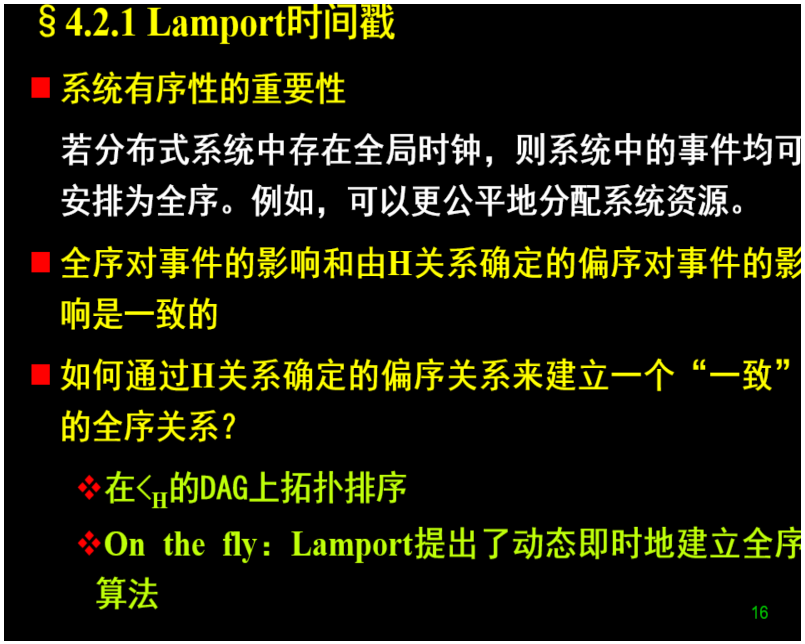
id-标识符

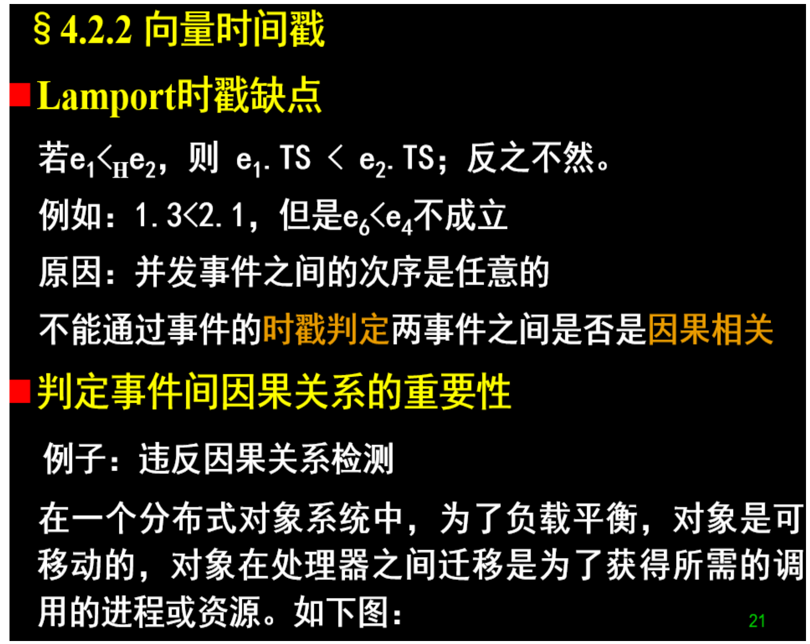
l-阶段数

hop-跳步计数器：初值为0，结点转发probe消息时加1.

若一结点收到的probe消息时，hop值为2^l，则它是2^l邻居中最后一个处理器。若此时msg未被没收也不能向前转发，而应该是向后发回reply消息。

向量时间戳





- 并发事件
- 因果相关事件

（1，1，0，4）
（1，1，2，3）
（2，5，0，0）
（3，6，4，3）

同步环选举算法

§3.4.2 有限制算法的下界Ω(nlgn)

- 同步的下界不可能从异步的下界导出

  因为上节中的算法表明：同步模型中的附加假定是必不可少的。

- 同步的下界对于非均匀和均匀算法均成立，但异步的下界只对均匀算法成立。
- 但是从同步导出的异步结果是正确的，并且提供了一个非均匀算法的异步下界。

异步通信模型中领导者选举问题
所需的消息数下界为Ω(nlgn)且
算法不依赖于比较的或者限时的

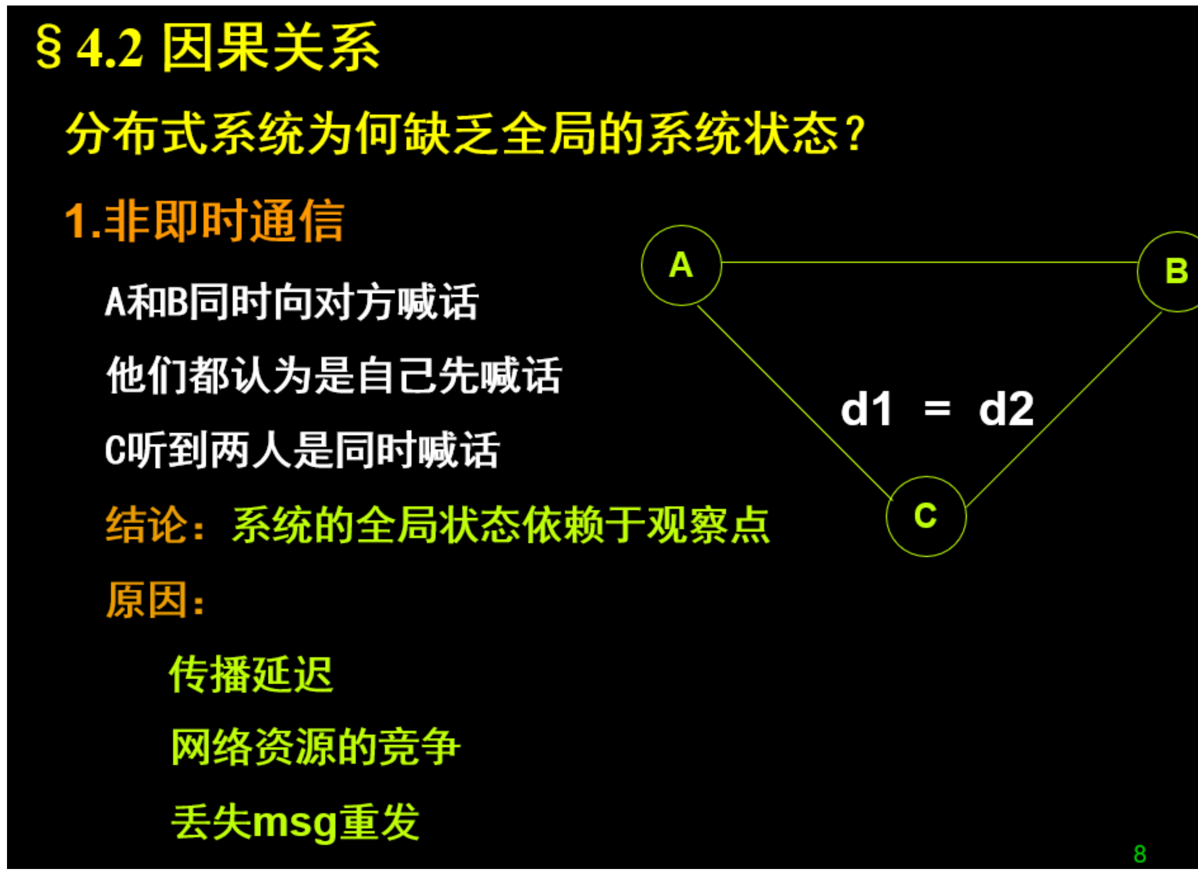
因果关系

分布式系统为何缺乏全局的系统状态？

1、非即时通信

2、相对性影响

3、中断



目　录

中国科学技术大学
University of Science and Technology of China

- 习题中出现的问题
  - 第一次作业
  - 第二次作业
  - 第三次作业
  - 第四次作业
- 习题解答
- 期末考试

概率算法作业的问题

- **EX 1.** 的答案是：$2\sqrt{2}$
  - $\frac{\sqrt{2}}{2}$ ✕
  - $4\sqrt{2}$ ✕
  - 2.8 ✕
  - 估计出来的 pi 值为 $2\sqrt{2}$
  - $8^{0.5}$ 最好写成 $2\sqrt{2}$

- **EX 3.** 大部分同学的积分区间都设置为了正数，最好考虑一个更加普遍的情形。
- **EX 4.** 集合计数有同学算出的结果波动特别大，最好运行多次求平均值。

概率算法作业的问题

- **EX 1. 八皇后问题-证明**
  - $$\frac{1}{i} \times \prod_{i}^{i\to N} \frac{i}{i+1} = \frac{1}{N}$$ ✗
  -

- **EX 2. 寻找最优的 StepVegas**
  - 应该给出**实验结果**，然后分析最优的 StepVegas 是多少（最好给出**理由**为什么是最优的，比如说总的时间或者搜索节点数最小）

概率算法作业的问题

➢ EX 1. 素性检验

- 很多同学都是简单地给出了实验结果，最好对实验结果进行总结得出结论，然后说明为什么会出现这样的结果。

分布式算法作业问题

中国科学技术大学 University of Science and Technology of China

- **EX 1. convergecast 时间复杂度分析**
  - 时间复杂度要带符号 “O()”；
  - 题目的答案 O(d)，d表示树的深度；
  - 很多同学写成了O(n)，n 是节点的个数。

- **EX 2. 可达←→其parent变量赋值**
  - 这里要证的是当且仅当，需要从充分性和必要性两个方面来证明，很多同学只证明了充分性。

分布式算法作业问题

中国科学技术大学 University of Science and Technology of China

- **EX 3. 证明 Alg2.3 →以 pr 为根的DFS树**
  - 有些同学直接证深度优先性;
  - 应该先证明构造了一棵树（从连通性和无环性说明）;
  - 然后再证明该生成树是 DFS 树。
- **EX 4. 证明Alg 2.3的时间复杂度为O(m)**
  - 注意这里的证明需要分模型（同步和异步）讨论，有些同学只考虑了同步模型下的证明。
- **EX 5. 修改算法Alg 2.3**
  - 有些同学只是简单地描述方法的思路，且描述得不够清楚，最好结合算法的伪代码进行说明 。
  - 最好分析一下设计的算法的时间复杂度为什么是O(n)

习题解答

Ex 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。

证明：

对于任意 $m \in Z$ 满足 $1 \le m \le n_b$，第 m 个位置被选中的概率等于

$$\frac{1}{m} \times \frac{m}{m+1} \times \frac{m+1}{m+2} \times \cdots \times \frac{n_b - 1}{n_b} = \frac{1}{n_b}$$

故对于 $(k+1)th$ 皇后，若有个开放位置，则每个位置被选中的概率都是 $\dfrac{1}{n_b}$。

Ex2.4 证明 Alg2.3 的时间复杂性为 O(m)。

解：

同步模型：每一轮中，根据算法，有且只有一个消息(M or Parent or Reject)在传输，从算法的第 6 、14、16、20、25 行发送消息的语句中可以发现：消息只发往一个处理器结点，除根结点外，所有的处理器都是收到消息后才被激活，所以，不存在多个处理器在同一轮发送消息的情况，所以时间复杂度与消息复杂度一致。

异步模型：在一个时刻内至多有一个消息在传输，因此，时间复杂度也与消息复杂度一致。消息复杂度：对任一边，可能传输的消息最多有 4 个，即 2 个 M ，2 个相应 M 的消息（Parent or Reject），所以消息复杂度为 O(m)

综上分析，该算法的时间复杂度为 O(m)。

考试安排

- **考试形式**

  闭卷考试，不允许将任何电子设备、书籍、资料带入考场

- **范围**

  概率算法（50-60%）、分布式算法（40-50%）

- **考试时间**

  7月20日9：30-11：30

- **题型分布**

  选择题 （10 *3）+ 计算与简答题（4*10）+ 算法设计题（2*15）

祝大家考试顺利！