Alogrithm exercise

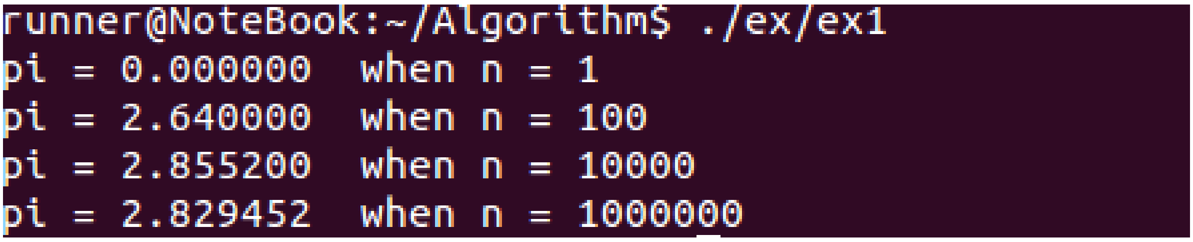
ywjia@mail.ustc.edu.cn
SA14011066
贾亚伟

2014-10-23

概率算法习题

1 ex1

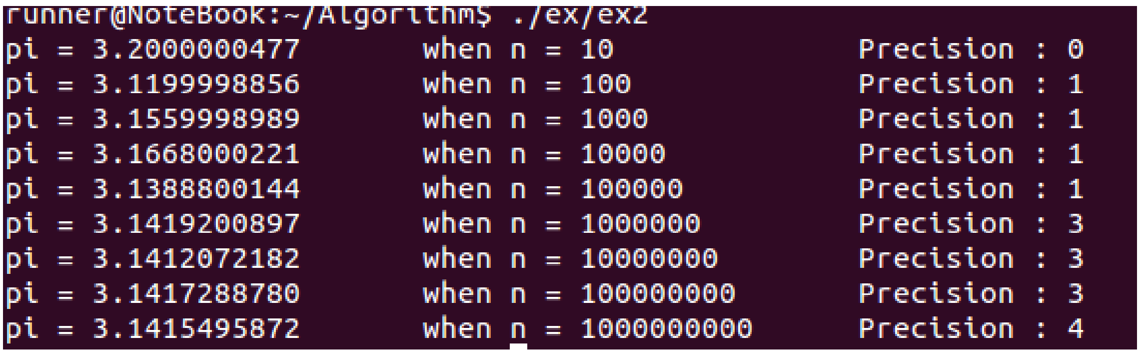
程序**ex1.c** 见 **ex** 文件目录
result as follows:



```
runner@NoteBook:~/Algorithm$ ./ex/ex1
pi = 0.000000  when n = 1
pi = 2.640000  when n = 100
pi = 2.855200  when n = 10000
pi = 2.829452  when n = 1000000
```

2 ex2

程序ex2.c 见 ex 文件目录
result as follows:



```
runner@NoteBook:~/Algorithm$ ./ex/ex2
pi = 3.2000000477       when n = 10             Precision : 0
pi = 3.1199998856       when n = 100            Precision : 1
pi = 3.1559998989       when n = 1000           Precision : 1
pi = 3.1668000221       when n = 10000          Precision : 1
pi = 3.1388800144       when n = 100000         Precision : 1
pi = 3.1419200897       when n = 1000000        Precision : 3
pi = 3.1412072182       when n = 10000000       Precision : 3
pi = 3.1417288780       when n = 100000000      Precision : 3
pi = 3.1415495872       when n = 1000000000     Precision : 4
```

3 ex3

问题:
calculate the integer f: [a, b] → [c, d]
here set f(x) = sin(x).
程序ex3.c 见 ex 文件目录
result as follows:

```
runner@NoteBook:~/Algorithm$ ./ex/ex3
please input a, b, c ,d ,n value:
 0 1 -1 1 2
Integer is:  2.000000
```

4 ex4

Question:
Assume ϵ, δ is constant within $(0, 1)$, prove:
If I is the correct value of $\int_0^1 f(x)dx$, and h is the return value of alogrithm "HitorMiss", Then $Prob[|h - I| < \epsilon] \geq 1 - \delta$ when $n \geq I(1-I)/\epsilon^2\delta$.
Prove:
Assume we hit the area n times and there are k points scatterd under the area $f(x)$.Apparently, the random X that the number of points scattered under $f(x)$ is Binomial Distribution, that is $P_r(X = k) = C_n^k p^k (1-p)^{n-k}$ then, $E(X) = np, Var(X) = np(1-p)$.
the h that HitOrMiss return is k/n, So $h = X/n, E(h) = E(X/n) = p = k/n = I, Var(h) = Var(X/n) = \frac{p(1-p)}{n} = \delta^2$.
According to Chebyshev's inequality $Pr(|h - I| \leq \epsilon) \geq 1 - \delta^2/\epsilon^2$ And according to the cond: $n \geq I(1-I)/\epsilon^2\delta$, So we have $Pr(|h - I| \leq \epsilon) \geq 1 - \frac{p(1-p)}{\epsilon^2 n} \geq 1 - \frac{p(1-p)}{\epsilon^2}\frac{\epsilon^2\delta}{I(1-I)} = 1 - \delta$.
Q.E.D

5 ex5

程序ex5.py 见 ex 文件目录
result as follows:

```
runner@NoteBook:~/Algorithm$ ./ex/ex5.py
the card of set is:
116176.48388236528
```

分析:
实验结果表明，当要估值的集合的基越大，也就是n 值越大 算法对n的估值越准确。

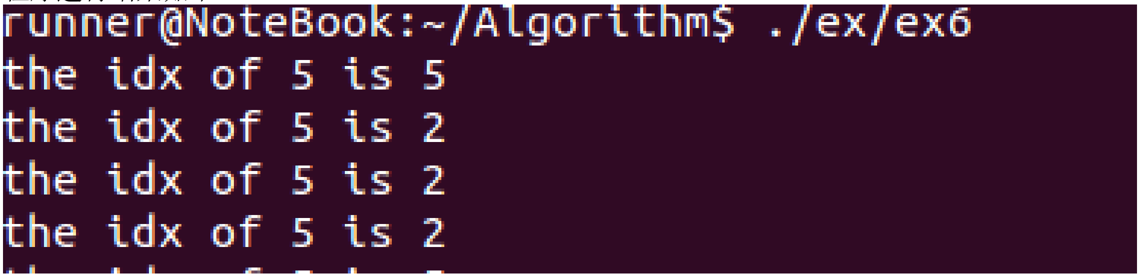
6 ex6

问题: 写一Sherwood算法C 与算法A，B，D比较，并给出实验结果。

答案:

程序ex6.c 见 ex 文件夹。

程序运行结果如下：



7 ex7

问题: 证明当放置第k+1个皇后时， 若有多个位置是开放的， 则算法QueensLV选中其中一位置的概率相等。

证： 假设在放置k+1个皇后时，有nb个位置可以放置， 并假设选中了第i个位置， 则其概率$p(i) = \frac{1}{i} \times \frac{i}{i+11} \times \cdots \times \frac{nb-1}{nb} = \frac{1}{nb}$ 即选中其中一个位置的概率均相等， 为$\frac{1}{nb}$。

8 ex8

写一算法，求n=12-20时最优的StepVegas值。

算法如下:

Algorithm 1 求StepVegas最优值

```
procedure OptimalStepVegas
    mintime ← ∞
    k ← 0
    for from n = 12 to 20 do
        for from k = 0 to 20 do
            time ← QueensLv(n, success, k)
            if time < mintime then
                mintime ← time
                stepVegas ← k
        Print TheCountofQueens : n, OptimalStepVegasValueis : stepVegas
```

9 ex9

问题：打印10000 以内的素数并与确定性算法相比较,并给出100 10000以内错误的比 例。 解:

程序ex9.c 见ex文件目录， 程序运行结果如下(部分）:

```
runner@NoteBook:~/Algorithm$ ./ex/ex9
2       3       5       7       11      13      17      29      41      97      1
93      257     769
```

错误比例为：(1204-845）/1204 = 29.5%
额，错误率有点高，但找了好久没找到原因

近似算法习题

10 完善证明

证： 设最优调度使得每台机器恰有2个作业：Ji和Jj，则必有i⩽m,j>m。否则若某最优调度O有i,j⩽m，则定有某台机器上有Js和Jt，使得s,t¿m. ∵Pi,Pj⩾Ps,Pt，交换Pj和Pt，则 Pi+Pt⩽Pi+Pj Ps+Pj⩽Pi+Pj 交换后的调度O’ 的最迟完成时间只可能减少，故O’ 也是最优调度。对于i,j¿m可类似证明。 ∴必有最优调度使J1,...,Jm分别分配到M1,⋯,Mm上，当将Jm+1,...,J2m分配到M台机器上时，LPT是将长时间的作业分配到轻负载上，必与该最优调度结果相同。

分布式算法习题

11 ex2.1

问题:分析在同步和异步模型下,convergecast 算法的时间复杂性。
答： 1) 在同步模型中 最坏情况下,算法执行的每一轮中只有一个 msg 传递,而此时生成树汇聚最大 值的算法最多执行 $n-1$ 轮(即生成树中除了末端节点每一个节点只有一个子节 点),也就是说同步情况下时间复杂度为 $O(n-1)$
(2) 在异步模中 在异步模型的汇集算法的每个容许执行中,树中每个距离 p_r为 t 的处理器至多 在时刻 t 接收消息 M,因此对于每个节点而言,它到它所有子节点中 t 最大的 路径决定了它本身时间花费。因此在最坏情况下,仍应该是同步模型下的最坏 情况,即生成树中除了末端节点每一个节点只有一个子节点,此时时间复杂度 仍为 $O(n-1)$

12 ex2.2

问题： G 里一结点从 pr 可达当且仅当它曾设置过自己的 parent 变量。
证:
必要性： 因为图 G 是由 parent 和 children 确定的静态图,任一节点在收到 M 后才会加入到图 中。即可达节点收到过 M,执行了算法 2.2 的第五行。由于是容许执行的,所以第 7 行(parent:=j)也会执行。
充分性： 一节点设置过自己的 parent 变量,则其从 p_r 可达。 若算法 2.2 的第 7 行执行过了,因为是容许执行,则必然有第 5 行也执行过了。即节 点收到过 M。而 M 又是从 p_r 发出的,所以该节点是从 p_r 可达的。

13 ex2.3

问题： 证明 Alg2.3 构造一棵以 Pr 为根的 DFS 树。
证：
（1）算法 2.3 构造的图 G 必然是连通的。否则,设 G 存在邻居节点 p_j 和 p_i。p_j 从 p_r 可 达,但 p_i 从 p_r 是不可达的。 则:

1) p_i 的 parent 为空;
2) p_i 不为 p_j 的 child.
因为: G 里一结点从 p_r 可达当且仅当它曾设置过自己的 parent 变量。 所以:
1) p_j 的 parent 必然设置过了;
2) p_i 的 parent 为 nill;
3) p_i 属于 p_j 的 unexplored 集合。
而算法的第 11 和 14 行决定了 p_j 会向 p_i 发送 M,使得 p_i 的 parent 成为 p_j,p_i 成为 p_j 的 child。 这与假设的结果矛盾。故p_i 必然也是从 p_r可达的。
（2）算法 2.3 构造的图 G 必然是无环的。否则设 G 中有一个环, p_1,p_2,...,p_i, p_1。令 p_1 是 该环中最早接收到 M 的节点。则 p_i 是从 p_1 可达的,且 p_1 的 parent 是 p_i,p_1 是 p_i 的 child。 而 p_i 在收到 M 后,向 p_1 发送 M。因为 p_1 的 parent 已经不为空,所以 p_1 收到来自 p_i 的 M 时,根据第 16 行代码,p_1 会向 p_i 放回一个$< reject >$信息,不会将 p_i 设为 parent。而 p_i 未收到 p_1 返回的¡parent¿信息,也不会将 p_1 设为 child。与前面的出 的结果矛盾。 故 G 是无环的。
(3) 图 G 是一棵 DFS 树。只需证明在有子结点与兄弟结点未访问时,子结点总是先加入 树中。 设有节点 p_1,p_2 和 p_3。p_2 和 p_3 是 p_1 的直接相邻节点。p_1 在第 12 14 行中先选择 向 p_2 发送 M,则 p_1 当且仅当 p_2 向其返回一个$< parent >$(第 17 行,第 22 行)时才 有可能向 p_3 发送 M。 而 p_2 仅在其向所有的相邻节点发送过 M 后才会向 p_1 返回$< parent >$ (第 19 21 行)。 所以 p_2 的子节点是永远先于 p_3 加入树中的,即 G 是 DFS 树。

14 ex2.4

问题：证明 Alg2.3 的时间复杂性为 O(m)
证明：
1):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。
(2)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,该算法的时间复杂度为 O(m)

15 ex2.5

问题：修改 Alg2.3 获得一新算法,使构造 DFS 树的时间复杂性为 O(n)。
解：
(1)在每个处理器中维护一本地变量,同时添加一消息类型,在处理器 Pi 转发 M 时,发 送消息 N 通知其余的为访问过的邻居,这样其邻居在转发 M 时便不会向 Pi 转发。
(2)在消息 M 和¡parent¿中维护一发送数组,记录已经转发过 M 的处理器名称。两种方式都是避免向已转发过 M 的处理器发送消息 M,这样 DFS 树外的边不再耗时, 时间。 复杂度也降为 O(n)

16 ex3.1

问题：证明同步环系统中不存在匿名的、一致性的领导者选举算法。
解：
在匿名系统中,每个处理器在系统中具有相同的状态机。由 Lemma3.1 可知,设算法 A 是使 环上某个处理器为 leader 的算法。因为环是同步的,且只有一种初始配置。在每轮里,各处 理器均发出同样的 message,所以在各轮里各个处理器接收到相同的 message,则状态改变也 相同。所以所有处理要么同为 leader,要么同时不为 leader。 故同步环系统中匿名的、一致性的领导者选举算法的算法是不存在的。

17 ex3.2

问题：证明异步环系统中不存在匿名的领导者选举算法。
解：
每个处理器的初始状态和状态机相同,除了接收消息的时间可能不同外,接收到的消息序列 也相同。所以最终处理器的状态也是一致的。由于处理器处理一条消息至多需要 1 单位时间,若某时刻某个处理器宣布自己是 leader,则在有限时间内,其它处理器也会宣布自己是 leader。故异步环系统中匿名的领导者选举算法是不存在的。

18 ex3.9

问题：若将环 R^{rev} 划分为长度为 j(j 是 2 的方幂)的连续片段,则所有这些片段是次序等价的.
解：
对一个整数$P, (0 \leq P \leq n-1)$ 可以表示为： $P = \sum_{i=1}^{m} a_i 2^{i-1}$, 其中 $m = \log n$, 则有$rev(P) = \sum_{i=1}^{m} a_i 2^{i-1}$
设 P, Q 在同一个片段上, P_1, Q_1在同一片段上， 且设这两个片段是相邻， 由模运算的加法可得:
$P_1 = P + l$
$Q_1 = Q + l$ 式中l表示片段的长度， $l = 2^k$. 另:

$$P = \sum_{i=1}^{m} a_i 2^{i-1}$$

$$Q = \sum_{i=1}^{m} a_i 2^{i-1}$$

且 P,Q 在同一片段上有 $|P - Q| \leq 1 = 2^k$ 所以存在 $r(0 \leq r \leq k)$, 满足$a_r \neq b_r$. 否则， $|P - Q| \ \geq 1$. 这与P, Q 在同一个片段上矛盾。
设$s = minr$, 则根据 rev(P), rev(Q)的表示方法可得:

$$sign()rev(P) - rev(Q) = sign(a - b)$$

而

$$P_1 = P + 1 = \sum_{i=1}^{m} a_i 2^{i-1} + 2^k$$

$$Q_1 = Q + 1 = \sum_{i=1}^{m} a_i 2^{i-1} + 2^k$$

显然， P 与P_1 的前k位相同，Q 与Q_1 的前k位相同。 由$0 \leq s \leq k$ 得

$$sign()rev(P) - rev(Q) = sign(a - b)$$

这两个相邻片段是序等价的,根据等价的传递关系,可得所有的片段都是次序等价的。