

MAC 115 – Introdução à Computação – IF Noturno

CARREIRAS LONGAS

Exercício-Programa 2 (EP2) Data de entrega: **13/10/2014 (23:55)**

1 Simulação de cara-ou-coroa

O programa `ep2a.c` abaixo simula uma seqüência de n lances de uma moeda que tem probabilidade p de dar cara.

```
/* ep2a.c */
#include <stdio.h>
#include <stdlib.h>

int Be(double p)
{ return rand()/(1.0 + RAND_MAX) < p; }

int main()
{
    int seed, n;
    double p;

    scanf("%d %lf %d", &seed, &p, &n);

    srand(seed);
    while (n-- > 0)
        printf("%d", Be(p));

    printf("\n");

    return 0;
}
```

Quando executado com entrada `271828 .3 200`, a saída pode ser, por exemplo, como segue:

```
10001110000000001010010000001100110100011001011110000010111110010000011001100101
0010000100000110110011000000000000010100000000001001010000110100101100000100100
00011101000101101001000100101010010110
```

Os 1 representam “cara” e os 0 representam “coroa”. (A saída acima foi editada para caber na página: uma mudança de linha foi inserida para que haja no máximo 80 caracteres por linha (há 80 + 80 + 40 caracteres acima).) Note que, em seu sistema, a saída pode ser uma seqüência diferente de 0s e 1s, pois o comportamento de `rand()` depende do sistema.

Suponha que você queira *contar o número de caras* geradas na simulação. Como você poderia alterar `ep2a.c` para fazer isso? Uma solução é alterar a função `main()` de `ep2a.c` para a seguinte versão:

```
int main()
{
    int seed, n, x, U = 0;
    double p;

    scanf("%d %lf %d", &seed, &p, &n);

    srand(seed);
    while (n-- > 0) {
        x = Be(p);
        if (x == 1) U++;
        printf("%d", x);
    }
    printf("\n");
    printf("Numero de 1s: %d\n", U);

    return 0;
}
```

2 Carreiras

Dada uma seqüência s de 0s e 1s, uma *carreira* de 1s em s é um segmento não-vazio maximal de 1s em s (um segmento que não tem um 1 à esquerda e não tem um 1 à direita). Por exemplo, há 47 carreiras de 1s na seqüência de comprimento 200 acima. Note que essas carreiras têm comprimentos variados. De fato, as mais curtas têm comprimento 1 e há exatamente uma de comprimento máximo, que tem comprimento 5. Vamos denotar o comprimento máximo de uma carreira de 1s em uma seqüência s por $M(s)$. Assim, se s é a seqüência de comprimento 200 acima, temos $M(s) = 5$. Denotemos por $N(s)$ o número de carreiras de 1s em s . Se s é a seqüência no exemplo acima, temos $N(s) = 47$.

Neste EP, você deve desenvolver um algoritmo que determina $N(s)$ e $M(s)$ para uma seqüência s de 0s e 1s gerada ao acaso, como em `ep2a.c`. Seu EP2 deve ser uma modificação adequada de `ep2a.c`.

2.1 Resultados teóricos

Prova-se que uma seqüência aleatória s de 0s e 1s de comprimento n com probabilidade p de 1s tem $M(s)$ em torno de

$$\log_{1/p}(n(1-p)) + \gamma/\log(1/p) - 1/2, \quad (1)$$

onde $\gamma = 0.57721566490153286060\dots$ é a constante de Euler–Mascheroni.

2.2 Experimentos computacionais

Note que, com o algoritmo que você desenvolverá neste EP, você poderá fazer experimentos computacionais para verificar a asserção na Seção 2.1: você pode experimentar gerar várias seqüências aleatórias s , de comprimentos variados, com valores de p variados, para verificar se $M(s)$ é de fato próximo à expressão dada em (1).

3 Seu EP

Escreva um programa, digamos `ep2.c`, como segue. Seu `ep2` deve receber como entrada uma “semente” $seed$, uma probabilidade p , um comprimento n e um inteiro positivo T . Ao ser executado com entrada $seed$, p , n e T , seu programa deve gerar T seqüências aleatórias de 0s e 1s, digamos s_1, \dots, s_T . Essas seqüências devem ter comprimento n , com 1s gerados com probabilidade p . Para cada s_i ($1 \leq i \leq T$), seu programa deve determinar $N(s_i)$ e $M(s_i)$. Note que seu programa examinará $N(s_1) + \dots + N(s_T)$ carreiras de 1s. Seja l o *comprimento médio* dessas carreiras de 1s.

Seu programa deve ter como saída o valor *médio* dos $N(s_i)$ e $M(s_i)$. Seu programa deve também ter como saída l , o comprimento médio das carreiras de 1s encontradas. Finalmente, seu `ep2` deve também imprimir o valor da expressão em (1).

Execute seu programa para valores grandes de T e n moderadamente grandes. Experimente vários valores de p . Compare o valor médio dos $M(s_i)$ e o valor da expressão (1). Você teria uma conjectura de como l depende de p ?

AVISOS IMPORTANTES

O EP deve ser feito individualmente!

Instruções sobre a entrega dos exercícios-programa

Todo exercício-programa deve ter o seguinte cabeçalho:

```
/* **** */
/* Fulano de Tal (seu nome) ..... No. USP:..... */
/* Exercício-Programa EPx */
/* Curso ..... - Turma ... - Prof. xxxxxxxxxxxxxxxxxxxx */
/* */
/* Compilador/plataforma: ..... */
/* **** */
```

- 1) O exercício-programa é **ESTRITAMENTE INDIVIDUAL**. Exercícios copiados (com ou sem eventuais disfarces) receberão nota **ZERO**.
- 2) **EXERCÍCIOS ATRASADOS NÃO SERÃO ACEITOS**.

- 3) Exercícios com erros de sintaxe não serão considerados.
- 4) É muito importante que seu programa tenha comentários e esteja bem indentado, ou seja, digitado de maneira a ressaltar a estrutura de subordinação dos comandos do programa (conforme visto em aula).
- 5) A saída do programa deve seguir as especificações dadas no enunciado. Pergunte ao professor e ao monitor em caso de dúvida.
- 6) A entrega do seu programa deve ser feita usando-se o sistema PACA. Para isso, você deve criar uma conta no sistema PACA, entrando em <http://paca.ime.usp.br>.
Se você ainda não possui uma conta no PACA, clique em *acesso* (no canto superior direito da página do PACA) e crie uma conta para você, utilizando obrigatoriamente o seu número USP. Tenha muito cuidado com a sua senha.
- 7) Para sua segurança, **GUARDE** com você **ATÉ** o **CADASTRO DE SUA NOTA FINAL**, uma cópia de cada programa entregue. Se, por algum motivo, o professor solicitar uma cópia e você não tiver, você poderá ser prejudicado.
- 8) As observações (4) a (6) influirão no critério de avaliação dos exercícios-programa.

Comece a fazer os exercícios-programa bem cedo!
O laboratório pode ficar muito cheio nas vésperas da data de entrega!