

Aula do CCM do dia 9/10/2013

Leonardo Nagami Coregliano

9 de outubro de 2013

Observação. Os exercícios numerados abaixo são do livro [1] e a sua numeração foi preservada. Alguns exercícios foram levemente alterados.

Exercício (1.4.37. Coeficientes binomiais (alterado)). *Escreva um programa que, dado um natural N , escreva as $N + 1$ primeiras linhas do triângulo de Pascal.*

Exercício (1.4.14). *Escreva um programa que recebe um inteiro N da linha de comando e cria uma matriz $N \times N$ booleana $a[][]$ tal que $a[i][j]$ é true se $i+1$ e $j+1$ são relativamente primos e é false caso contrário.*

Exercício (1.4.20 (EP). Simulação de dados). *O seguinte código computa a distribuição de probabilidade exata da soma de dois dados honestos:*

```
double[] dist = new double[13];
for (int i = 1; i <= 6; i++)
    for (int j = 1; j <= 6; j++)
        dist[i+j] += 1.0;
for (int k = 1; k <= 12; k++)
    dist[k] /= 36.0;
```

O valor $dist[k]$ é a probabilidade de que os dados somem k . Execute experimentos para validar esse cálculo simulando N jogadas de dados, acompanhando as frequências de ocorrência de cada valor de soma. Quão grande N tem de ser para que os resultados empíricos batam com os calculados com precisão até a terceira casa decimal?

Exercício (1.4.21 (EP). Plateau mais longo). *Dado um vetor de inteiros, determine o comprimento e a localização da maior sequência contígua de valores iguais onde os valores dos elementos imediatamente antes e depois dessa sequência não são maiores.*

Exercício (1.4.24. Mínimos em permutações (alterado)). *Escreva um programa que, dado um vetor a de inteiros, calcula quantos mínimos da esquerda para a direita há em a (o número de elementos que são menores do que todos os que aparecem à esquerda no vetor).*

Exercício (Ordenação). *Faça um programa que recebe uma sequência de inteiros nos argumentos da linha de comando e imprime-os em ordem crescente.*

Dica. Um dos algoritmos mais simples de ordenação consiste em repetidamente procurar o i -ésimo menor valor e colocá-lo na i -ésima posição.

Exercício (Embaralhando um vetor). *Escreva um trecho de código que recebe um vetor $a[]$ e transforma-o em uma permutação escolhida uniformemente ao acaso.*

Exercício (1.4.23. Embaralhada ruim (adaptado)). *Prove que o algoritmo abaixo não gera uma permutação escolhida uniformemente ao acaso de um vetor $a[N]$.*

```
for (int i = 0; i < N; ++i) {
    int r = (int)(N * Math.random());
    int aux = a[r];
    a[r] = a[i];
    a[i] = aux;
}
```

Exercício (1.4.30. Campo Minado). Escreva um programa que recebe 3 argumentos na linha de comando M , N e p e produz uma matriz $M \times N$ booleana, onde cada entrada é ocupada com probabilidade p (independentemente). No jogo de Campo Minado, células ocupadas representam bombas e células vazias representam células seguras. Imprima a matriz usando um asterisco para bombas e um ponto para células seguras. Em seguida, substitua cada célula vazia com o número de bombas vizinhas (aos lados e diagonais) e imprima o resultado.

```
* * . . . * * 1 0 0
. . . . . 3 3 2 0 0
. * . . . 1 * 1 0 0
```

Tente escrever seu código de forma que ele tenha poucos casos especiais a serem tratados usando uma matriz $(M + 2) \times (N + 2)$.

Extra: e se, em vez de p , recebêssemos um inteiro $k \leq MN$ e quiséssemos produzir um Campo Minado com k minas uniformemente ao acaso?

Exercício (Multiplicação de matrizes). Escreva um trecho de código que, dadas duas matrizes de inteiros $\mathbf{a}[][]$ e $\mathbf{b}[][]$ tais que o número de colunas de \mathbf{a} é o mesmo que o número de linhas de \mathbf{b} , calcula o produto dessas matrizes.

Exercício (1.4.15). Escreva um trecho de código que, dadas duas matrizes de booleanos $\mathbf{a}[][]$ e $\mathbf{b}[][]$ tais que o número de colunas de \mathbf{a} é o mesmo que o número de linhas de \mathbf{b} , calcula o produto dessas matrizes usando “ou” no lugar da soma de elementos e “e” no lugar da multiplicação de elementos.

Exercício (1.4.25. Inversas de permutações). Escreva um programa que, dada uma permutação \mathbf{a} , calcula sua inversa (i.e., calcula uma permutação \mathbf{b} tal que, para todo i , temos $\mathbf{a}[\mathbf{b}[i]] = i$).

Exercício (1.4.26. Matriz de Hadamard). A matriz de Hadamard $H(N)$ de ordem $N \times N$ é uma matriz booleana com a propriedade de que quaisquer duas linhas distintas diferem em exatamente $N/2$ posições. A matriz $H(1)$ é a matriz com uma única entrada `true` e, para $N \geq 1$, a matriz $H(2N)$ pode ser obtida a partir de $H(N)$ alinhando quatro cópias dela e invertendo as entradas da cópia inferior direita.

Escreva um programa que, dado um natural M , imprime $H(2^M)$.

Referências

- [1] R. Sedgewick and K.D. Wayne, *Introduction to programming in java: an interdisciplinary approach*, Pearson Addison-Wesley, 2008. (document)