

**Projeto e Análise de Algoritmos – Simulação da Primeira Prova – 15/09/2014**

1. (2.0) Abaixo estão duas soluções para cada um dos seguintes problemas (2-sum e 3-sum): a partir de um vetor de N números inteiros distintos, determinar o número de pares, ou triplas, que possuem soma igual a zero. Explique cada solução apresentada, indicando a complexidade de cada algoritmo. Qual solução é mais eficiente para cada problema?

<pre> 01 int method1-2sum( int a[], int N ) { 02     int cnt = 0; 03     for (int i = 0; i &lt; N; i++) 04         for (int j = i+1; j &lt; N; j++) 05             if (a[i] + a[j] == 0) 06                 cnt++; 07     return cnt; 08 }</pre>	<pre> 01 int method2-2sum( int a[], int N ) { 02     int cnt = 0; 03     sort( a, N ); 04     for (int i = 0; i &lt; N; i++) 05         if ( search( a, N, -a[i] ) &gt; i ) 06             cnt++; 07     return cnt; 08 }</pre>
<pre> 01 int method1-3sum( int a[], int N ) { 02     int cnt = 0; 03     for (int i = 0; i &lt; N; i++) 04         for (int j = i+1; j &lt; N; j++) 05             for (int k = j+1; k &lt; N; k++) 06                 if (a[i] + a[j] + a[k] == 0) 07                     cnt++; 08     return cnt; 09 }</pre>	<pre> 01 int method2-3sum( int a[], int N ) { 02     int cnt = 0; 03     sort( a, N ); 04     for (int i = 0; i &lt; N; i++) 05         for (int j = i+1; j &lt; N; j++) 06             if ( search( a, N, -a[i]-a[j] ) &gt; j ) 07                 cnt++; 08     return cnt; 09 }</pre>

OBS: O procedimento  $sort(a,b)$  ordena um vetor  $a$  de tamanho  $b$ . A função  $search(a, b, c)$  procura em um vetor  $a$  de tamanho  $b$  o elemento  $c$ , retornando o índice do elemento encontrado, ou -1 caso não o encontre.

2. (2.0) Abaixo estão duas implementações diferentes (POL-HORNER e POL) para calcular um polinômio de grau K. Responda as questões abaixo, justificando sua resposta:

$P(x) = \sum_{i=0}^K a_i x^i$ $P(x) = a_0 + x(a_1 + x(a_2 + \dots + x(a_{k-1} + xa_k) \dots))$	<b>POL-HORNER(x,k)</b> $y = 0$ for i = k downto 0 $y = a[i] + x*y$ return y	<b>POL(x,k)</b> $y = 0$ for i = 0 to k $y = y + a[i]*POT(x,i)$ return y	<b>POT(x,k)</b> $y = 1$ for i = 1 to k $y = y*x$ return y
--	---	---	---

- (a) Em termos da notação  $\Theta$ , qual o tempo de execução das funções POL-HORNER e POL?  
(b) Qual dos dois métodos (POL-HORNER e POL) é o melhor?

3. (4.5) Faça um programa que encontra um caminho válido em um labirinto de tamanho  $N \times M$ , considerando como início a posição (1,1) e fim a posição (N,M). Apenas quatro movimentos são permitidos, para cima, baixo, esquerda e direita. Documente as decisões que fizer e informe a complexidade da sua solução. A seguir temos um exemplo de descrição de um labirinto 7x5:

Matriz de entrada:

0	1	0	0	0
0	0	0	1	0
1	1	0	1	1
0	0	0	0	0
1	0	1	1	1
1	0	0	0	1
1	1	1	0	0

Labirinto:


4. (1.5) Forneça a estimativa mais apropriada para descrever a frequência das operações de cada um dos algoritmos abaixo:

<b>(a)</b> 01 for (i=0; i<n; i++) 02     for (j=0; j<i; j++) 03         if( vetor[i] > 0 ) 04             x = x + j;	<b>(b)</b> 01 for (i=1; i<n; i*=2) 02     x = x + i;	<b>(c)</b> 01 for (int n = k; n > 0; n /= 2) 02     for(int i = 0; i < k; i++) 03         sum++;
--	--	---