Segunda Avaliação

COMP0412 - 2024.2 - T02

- As funções da Questão 1 precisam ser implementadas, não basta o pseudo-código.
- Assuma que estão disponíveis as funções mergeSort, quickSort, insertionSort, quickSelect, binarySearch e convexHull.
  (caso não seja pseudo-código, pode usar qualquer implementação destas funções)

1. Dado um conjunto de pontos bi-dimensionais, apresentados em um arquivo de texto no seguinte formato: a linha 1 contém o total N de pontos, as linhas 2 a N+1 cotêm as coordenadas separadas por vírgulas de cada ponto. O separador da parte fracionária será o ponto.

Exemplo: Os conjunto de pontos {(1, 2), (0.5, 3), (2, 0.7), (3, 5)} seria apresentado no arquivo como

```
4
1, 2
0.5, 3
2, 0.7
3, 5
```

a) Implemente uma função para ler o conjunto de pontos e armazenar em dois vetores, um vetor X com as primeiras coordenadas e um vetor Y com as segundas coordenadas. (não vale ponto)

b) Implemente uma função poly(X, k) que retorna uma matriz A_k com (k+1) colunas de modo que os valores da coluna i são os valores de X elevados a (k + 1 - i). (não vale ponto)

Para os dados do exemplo anterior teríamos:

$$A_1 = \begin{vmatrix} 1 & 1 \\ 0.5 & 1 \\ 2 & 1 \\ 3 & 1 \end{vmatrix} \quad A_2 = \begin{vmatrix} 1 & 1 & 1 \\ 0.25 & 0.5 & 1 \\ 4 & 2 & 1 \\ 9 & 3 & 1 \end{vmatrix} \quad A_3 = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 0.125 & 0.25 & 0.5 & 1 \\ 16 & 4 & 2 & 1 \\ 27 & 9 & 3 & 1 \end{vmatrix}$$

c) Implemente uma função inv(M) para inverter uma matriz quadrada M.

d) Implemente uma função ajusta(X, Y, k) que retorne

$P = inv(A^T A) A^T Y$

onde A é dado por poly(X, k).
(você pode usar funções prontas para produto e transposição se quiser)

Os valores de P são os coeficientes do polinômio de grau k que melhor se ajustam ao conjunto de pontos segundo o critério de minimização da soma dos quadrados dos erros (SSE), onde os erros são dados por E = AP - Y.

Sugestões para incluir no retorno de sua função (precisa no próximo item):

- **Tempo do cálculo de P.**
- **SSE.**

e) Você receberá 3 arquivos de dados (enviados posteriormente), calcule o que se pede para cada arquivo e preencha a tabela abaixo:

	k	Tempo de Execução (cálculo de P)	SSE
Arquivo 1	1		
	2		
	10		
Arquivo 2	1		
	2		
	10		
Arquivo 3	1		
	2		
	10		

f) Qual a complexidade de tempo teórica do seu código para o cálculo de P em função de N (número de pontos) e K (ordem do polinômio)? Os resultados obtidos na tabela estão de acordo com o que era esperado?

2. **A estratégia de ajuste de polinômios descrita na Questão 1 pode ser estendida para o ajuste de qualquer combinação linear de funções. Se você deseja ajustar a curva $p_1f_1(x) + p_2f_2(x) + \ldots + p_kf_k(x) = y$ aos pontos, basta montar uma matriz A onde a coluna i contém os valores de $f_i(x)$ para cada ponto X dos dados e repetir o procedimento anterior para obter os valores dos parâmetros p_i no vetor P.**

A função $p_1*\exp(p_2*x) = y$ não é linear, pois o parâmetro p_2 está dentro da função exponencial e não é um simples multiplicador.

a) **Transforme este problema em um problema linear.**

b) **Escreva uma função ajustaExp(X, Y) que utiliza a função ajusta, da Questão 1, e transforma a saída P para obter p_1 e p_2 de $p_1*\exp(p_2*x) = y$.**

3. **Considere que temos uma lista de valores reais $x_1, \ldots, x_N$. O valor x (não necessariamente pertencente à lista) que minimiza a soma de $|x - x_i|$ para i de 1 a N é a mediana dos x_i (no caso de N par, a mediana é a média dos valores centrais).**

Considere agora que temos uma lista de pontos no plano cartesiano $p_1=(x_1, y_1), \ldots, p_N=(x_N, y_N)$. Queremos achar um ponto p=(x, y) (não necessariamente da lista) que minimiza a soma da distância Manhattan entre cada p_i e p. Essa distância é dada por $d(p, p_i) = |x - x_i| + |y - y_i|$.

a) **Crie uma função pol(P) que retorna o ponto p que resolve o problema do segundo parágrafo, onde $P = [p_1, p_2, \ldots, p_N]$.**

b) **Qual a complexidade de tempo da sua solução?**

4. **Dado um conjunto de pontos $P = [p_1, p_2, \ldots, p_N]$, onde $p_i = (x_i, y_i)$.**

a) **Proponha uma função inTri(P) que retorna verdadeiro se todos os pontos de P estão contidos em um triângulo formado por 3 dos pontos de P.**

b) **Qual a complexidade de inTri(P)?**

c) **Proponha uma função inCirc(P, r) que retorna verdadeiro se todos os pontos de P estão contidos em um círculo de raio r centrado na média dos pontos de P.**

d) **Qual a complexidade de inCirc(P, r)?**

5. **O método de Horner para cálculo do valor de um polinômio $p(x) = a_k x^N + a_{k-1}x^{N-1} + \ldots + a_1 x + a_0$ em um dado ponto c é bastante eficiente, se $A = [a_0, a_1, \ldots, a_k]$, podemos avaliar p(c) como segue:**

```
Horner(A[0..N], c)
  p ← A[N]
  para i de (N-1) até 0:
    p ← c * p + A[i]
  retorne p
```

No entanto, para o cálculo de c^N, ou seja, avaliar $q(x) = x^N$ no ponto c, este método torna-se simplesmente (N-1) multiplicações de c (e várias somas com 0, desnecessárias), que é o que faríamos por força bruta.

Um número natural N pode ser representado como $b_K 2^K + b_{K-1}2^{K-1} + \ldots + b_1 2 + b_0$, onde b_i são os bits da representação binária de N. Ou seja, N é o valor do polinômio $r(x) = b_K x^K + b_{K-1}x^{K-1} + \ldots + b_1 x + b_0$ avaliado no ponto 2. Portanto, temos que $c^N = c^{r(2)}$.

Aplicando Horner em r(2) temos:

```
p ← 1  (o bit mais significativo é sempre 1, se N > 0)
para i de (K-1) até 0:
  p ← 2p + b_i
retorne p
```

De modo que para $c^{r(2)}$ temos:

```
c^p ← c
para i de (K-1) até 0:
  c^p ← c^(2p + bi)
retorne c^p
```

Esta versão não vai funcionar do modo como está escrito, já que c^p deveria ser uma variável, para receber a atribuição, mas o valor do expoente é alterado na atribuição do laço.

a) **Utilize as identidades da potenciação para reescrever $c^{2p + bi}$ como operações sobre c^p, de modo que ele possa ser tratado como uma variável no pseudo-código acima. (não vale ponto)**

b) **Utilizando o resultado do item (a) e o pseudo-código acima, escreva um algoritmo expBin(c, B_N) que calcula c^N, onde c é um número e $B_N = [b_0, b_1, \ldots b_K]$ são os bits da representação binária de N.**

c) **Qual a complexidade de tempo de expBin(c, B_N) em função de N?**

6. Considere que A é um array contendo N strings de comprimentos variados, formadas exclusivamente por letras minúsculas e sem acentos.

a) Qual método de ordenação você utilizaria para ordenar A eficientemente? Qual a complexidade temporal do método escolhido para este problema?

b) Duas (ou mais) strings são anagramas se uma é uma permutação dos símbolos da(s) outra(s). Crie uma função anagroup(A) que retorna os anagramas contidos em A agrupados.

c) Qual a complexidade de anagroup(A)?

Exemplo:

Entrada: A = [cao, jaca, oca, cora, arar, ora, rara, caja, aco, aro]
Retorno: [[aco, cao, oca], [aro, ora], [arar, rara], [caja, jaca], [cora]]