

Java

Lista 1

Souza. Sara da Cunha M. Silva. Uanderson C.

Curso de Engenharia da Computação
Disciplina de Programação Paralela
Instituto Federal Fluminense

Maio, 2019

Exercícios

<https://github.com/saramonteiro/Java>

Java

Souza, Sara
da Cunha
M.,
Silva, Uan-
derson
C.

Exercício 1

Exercício 2

Exercício 3

Exercício 4

Exercício 5

Exercício 6

Exercício 7

Exercício 8

1 Exercício 1

2 Exercício 2

3 Exercício 3

4 Exercício 4

5 Exercício 5

6 Exercício 6

7 Exercício 7

8 Exercício 8

Enunciado

Java

Souza, Sara
da Cunha
M.,
Silva, Uan-
derson
C.

Exercício 1

Exercício 2

Exercício 3

Exercício 4

Exercício 5

Exercício 6

Exercício 7

Exercício 8

1) Considere a matriz $A = [a_{ij}]_{n \times m}$, onde $n = 4$ e $m = 5$, com números inteiros positivos gerados aleatoriamente de 1 até 20. Faça um algoritmo para gerar a matriz A e verificar se ela satisfaz a seguinte condição:

$$\min_{0 \leq j \leq m-1} \sum_{i=0}^{n-1} a_{ij} \leq \max_{0 \leq i \leq n-1} \prod_{j=0}^{m-1} a_{ij}$$

Crie um procedimento para gerar a matriz e uma função para realizar a verificação. De acordo com o retorno da função de verificação, deve-se imprimir na função main: "Condicao Satisfeita" ou "Condicao Nao Satisfeita".

Algoritmo

Java

Souza, Sara
da Cunha
M.,
Silva, Uan-
derson
C.

Exercício 1

Exercício 2

Exercício 3

Exercício 4

Exercício 5

Exercício 6

Exercício 7

Exercício 8

- Cria-se matriz 4 x 5
- Preenche-se matriz com valores randômicos
- Calcula-se a soma mínima entre as colunas usando 2 loops: 1 loop mantém a coluna e outro percorre a coluna incrementando as linhas.
- Calcula-se o produto máximo entre as linhas usando 2 loops: 1 loop mantém a linha e outro percorre a linha incrementando as colunas.
- Verifica-se a condição através da comparação entre os dois resultados anteriores.

Enunciado

Java

Souza, Sara
da Cunha
M.,
Silva, Uan-
derson
C.

Exercício 1

Exercício 2

Exercício 3

Exercício 4

Exercício 5

Exercício 6

Exercício 7

Exercício 8

2) Considere uma matriz M de ordem 3 de números inteiros armazenada no arquivo “MatrizM.txt”. Faça um algoritmo para ler esta matriz do arquivo e imprimir na tela se ela é ou não uma Matriz Ortogonal. Utilize três procedimentos: um para gerar a matriz M , outro para calcular a sua matriz transposta (M^T) e o terceiro para calcular a multiplicação $M \cdot M^T$.

Utilize também uma função para retornar se a matriz M é Ortogonal ou não. A impressão dessa informação tem que ser na função main.

Obs.: Se uma matriz quadrada M é uma matriz ortogonal, então $M \cdot M^T = I$, onde M^T é a matriz transposta de M e I a matriz identidade.

Algoritmo

Java

Souza, Sara
da Cunha
M.,
Silva, Uan-
derson
C.

Exercício 1

Exercício 2

Exercício 3

Exercício 4

Exercício 5

Exercício 6

Exercício 7

Exercício 8

- Cria-se a matriz M de ordem 3, a matriz M^t que recebe a transposta de M , e a matriz R .
- Preenche-se a matriz M com valores inteiros aleatórios.
- Calcula-se a transposta de M e a armazena em M^t .
- Calcula-se $M \cdot M^T$ e armazena o resultado na matriz R .
- É verificado se R é igual a matriz identidade.
- Se for ortogonal, a função de verificação retorna 0, senão, retorna 1.

Enunciado

Java

Souza, Sara
da Cunha
M.,
Silva, Uan-
derson
C.

Exercício 1

Exercício 2

Exercício 3

Exercício 4

Exercício 5

Exercício 6

Exercício 7

Exercício 8

3) Considere um vetor que armazena 10 números inteiros pares e 10 números inteiros ímpares todos embaralhados, ou seja, sem qualquer ordem preestabelecida. Faça um algoritmo para ler esse vetor do arquivo “Vetor.txt” e depois organizá-lo de modo que os números pares fiquem nas posições ímpares do vetor e os números ímpares fiquem nas posições pares do vetor. Crie dois procedimentos: um para preencher o vetor com os números do arquivo e o outro para organizá-lo. Obs.: Não é permitido utilizar qualquer outro vetor/matriz para auxiliar a organização.

Algoritmo

Java

Souza, Sara
da Cunha
M.,
Silva, Uan-
derson
C.

Exercício 1

Exercício 2

Exercício 3

Exercício 4

Exercício 5

Exercício 6

Exercício 7

Exercício 8

- Cria-se um vetor estático com 10 valores pares e 10 valores ímpares inteiros entre 0 a 20.
- Um loop externo tem a função de percorrer o vetor através de um índice (índice atual).
- A cada iteração é verificada a paridade do índice atual do vetor.
- Outro índice (índice móvel) percorre o vetor da posição atual em diante até encontrar um valor cuja paridade é oposta a do índice atual.
- Quando há esse encontro, o valor do vetor na posição do índice atual é trocada com o valor do vetor na posição do índice móvel.

Enunciado

Java

Souza, Sara
da Cunha
M.,
Silva, Uan-
derson
C.

Exercício 1

Exercício 2

Exercício 3

Exercício 4

Exercício 5

Exercício 6

Exercício 7

Exercício 8

4) Considere o arquivo “Numeros.txt” com 30 números inteiros. Faça um algoritmo para armazenar esses números em um vetor e depois ordenar este mesmo vetor de maneira não-decrescente. Utilize três procedimentos: um para preencher o vetor, outro para ordenar o vetor e um terceiro para imprimir o vetor antes e depois da ordenação.

Obs.: Não é permitido utilizar qualquer outro vetor/matriz para auxiliar a ordenação.

Algoritmo

Java

Souza, Sara
da Cunha
M.,
Silva, Uan-
derson
C.

Exercício 1

Exercício 2

Exercício 3

Exercício 4

Exercício 5

Exercício 6

Exercício 7

Exercício 8

- Cria-se um vetor de 30 posições.
- Preenche-se esse vetor com números inteiros aleatórios.
- Apresenta-se o vetor antes da ordenação.
- Ordena-se o vetor utilizando o algoritmo de ordenação Selection Sort.
- Apresenta-se o vetor após a ordenação.

Enunciado

Java

Souza, Sara
da Cunha
M.,
Silva, Uan-
derson
C.

Exercício 1

Exercício 2

Exercício 3

Exercício 4

Exercício 5

Exercício 6

Exercício 7

Exercício 8

5) Considere um número inteiro n com:

$(n \geq 0)$

lido pelo teclado. Faça um algoritmo recursivo para calcular o fatorial de n .

Algoritmo

Java

Souza, Sara
da Cunha
M.,
Silva, Uan-
derson
C.

Exercício 1

Exercício 2

Exercício 3

Exercício 4

Exercício 5

Exercício 6

Exercício 7

Exercício 8

- Programa aguarda usuário entrar com um valor para ser calculado o fatorial.
- Uma função recursiva recebe como único parâmetro o valor que se deseja calcular o fatorial. A função retorna um inteiro.
- Enquanto o parâmetro não chega a 1, a função decrementa o valor e chama a si própria.
- A primeira função a ser finalizada retorna um e inicia o desempilhamento.
- O valor retornado é sempre multiplicado pelo valor atual até finalizar a primeira função chamadora.

Enunciado

Java

Souza, Sara
da Cunha
M.,
Silva, Uan-
derson
C.

Exercício 1

Exercício 2

Exercício 3

Exercício 4

Exercício 5

Exercício 6

Exercício 7

Exercício 8

6) Considere dois números inteiros a e b ($b \geq 0$) lidos pelo teclado. Faça um algoritmo recursivo para calcular o valor de a^b .

Algoritmo

Java

Souza, Sara
da Cunha
M.,
Silva, Uan-
derson
C.

Exercício 1

Exercício 2

Exercício 3

Exercício 4

Exercício 5

Exercício 6

Exercício 7

Exercício 8

- São lidos pelo teclado o valor de a (base) e de b (expoente).
- A função recursiva recebe esses parâmetros.
- Se $b = 0$, a função retorna 1.
- Se $b = 1$, a função retorna o próprio a .
- Se o valor de b for maior que 1, a função retorna o valor de a multiplicado pela chamada da própria função mas com o valor de b decrementado.
- Quando a condição de parada é satisfeita, inicia-se o desempilhamento e ao final desse processo é obtido o valor de a^b .

Enunciado

Java

Souza, Sara
da Cunha
M.,
Silva, Uan-
derson
C.

Exercício 1

Exercício 2

Exercício 3

Exercício 4

Exercício 5

Exercício 6

Exercício 7

Exercício 8

7) Considere um vetor com 20 números naturais maiores do que 1 lidos pelo teclado. Faça um algoritmo recursivo que organize esse vetor de modo que os números compostos fiquem nas primeiras posições e os números que não são compostos nas últimas posições. Essa organização deve ser realizada sem utilizar qualquer estrutura de dados auxiliar. Crie e utilize dois procedimentos: um para preencher o vetor e outro recursivo para realizar a organização do vetor. Crie e utilize também uma função para retornar 1, se um número natural for composto, ou retornar 0, caso contrário. Obs. 1: Um número natural C é composto se ele tem mais de dois divisores naturais distintos. Obs. 2: Não é permitido utilizar qualquer outro vetor/matriz para auxiliar a ordenação.

Algoritmo

Java

Souza, Sara
da Cunha
M.,
Silva, Uan-
derson
C.

Exercício 1

Exercício 2

Exercício 3

Exercício 4

Exercício 5

Exercício 6

Exercício 7

Exercício 8

- A função recursiva recebe como parâmetro o vetor que deseja organizar, o início e o fim. Não há retorno.
- O critério de parada é se o vetor chegar ao fim e não houver mais o que ser comparado.
- A função possui três fluxos possíveis:
- O vetor chega ao fim. (Critério de parada).
- O conteúdo da posição atual (partida) do vetor é composto, a função é requisitada novamente atualizando a partida para posição seguinte. (Obs: quando essa função for desempilhada, não há mais nada a ser realizado).

Algoritmo

Java

Souza, Sara
da Cunha
M.,
Silva, Uan-
derson
C.

Exercício 1

Exercício 2

Exercício 3

Exercício 4

Exercício 5

Exercício 6

Exercício 7

Exercício 8

- O conteúdo da posição atual (partida) do vetor não é composto. Há a troca do conteúdo atual com o conteúdo da última posição do vetor. Isso garante que a última posição fica com um conteúdo não composto, dessa maneira, ele não precisa ser analisado no futuro e o fim é atualizado para um índice a menos e a função é chamada. Ao ser chamada a partida não muda, isso garante que se durante a troca outro número não composto ocupe a posição atual, ele será novamente trocado, até que receba um número composto, ou o fim chegue na partida indicando que não havia mais números compostos daquela posição em diante.

Enunciado

Java

Souza, Sara
da Cunha
M.,
Silva, Uan-
derson
C.

Exercício 1

Exercício 2

Exercício 3

Exercício 4

Exercício 5

Exercício 6

Exercício 7

Exercício 8

8) Considere o arquivo “Numeros.txt” com 20 números inteiros que devem ser armazenados em um vetor. Faça um algoritmo recursivo para imprimir o maior valor deste vetor.

Obs.: Não é permitido utilizar qualquer outro vetor/matriz para auxiliar.

Algoritmo

Java

Souza, Sara
da Cunha
M.,
Silva, Uan-
derson
C.

Exercício 1

Exercício 2

Exercício 3

Exercício 4

Exercício 5

Exercício 6

Exercício 7

Exercício 8

- Cria-se um vetor de 20 posições.
- Preenche-se esse vetor com números inteiros aleatórios.
- Mostra-se esse vetor antes da busca pelo maior número existente no mesmo.
- A função recursiva recebe esse vetor como parâmetro.
- O vetor é dividido ao meio e cada parte é enviada como parâmetro para uma chamada da mesma (somente o valor que representa a quantidade de posições é dividida. O vetor permanece o mesmo).

Algoritmo

Java

Souza, Sara
da Cunha
M.,
Silva, Uan-
derson
C.

Exercício 1

Exercício 2

Exercício 3

Exercício 4

Exercício 5

Exercício 6

Exercício 7

Exercício 8

- Esse passo se repete até que sobrem no máximo pares das posições (O valor que representa a quantidade de posições deve ser 2 ou 1).
- Ao final dessa divisão, os valores presentes nos pares são comparados entre si e o maior deles é retornado. E se uma posição ficar sem par, o valor presente nessa posição do vetor é retornado.
- Enquanto as chamadas recursivas são desempilhadas, os seus retornos são comparados par a par, onde o maior valor prevalece.
- Terminado esse processo, o maior valor presente no vetor é mostrado.