

Pontifícia Universidade Católica de Minas Gerais

Curso: Engenharia de Software

Disciplina: Algoritmos e Estruturas de Dados II

Professor: Rodrigo Richard Gomes

Entrega: 28/05/2019

DUPLA – 20 pontos

PARTE 1

Esse trabalho consiste em implementar e testar as operações de inserção e consulta em uma Árvore Binária de Pesquisa (ABP) e em uma Árvore AVL Em Java. Para isso você deve inserir elementos nessas árvores e medir o tempo para a inclusão. Você também deverá fazer a consulta de elementos nas árvores e medir o tempo total para as consultas. Tanto para a inserção quanto para a consulta, você deve utilizar os seguintes vetores:

- Vetor A : preenchido com os valores entre 1 e 10.000.000 (ou o maior tamanho possível)
- Vetor B : preenchido com os valores entre 10.000.000 e 1 (vetor invertido)
- Vetor C: preenchido aleatoriamente

Vetor A

1	2	3	4	5	6	...	9999999	10000000
0	1	2	3	4	5	...	9999998	9999999

Vetor B

10000000	9999999	9999998	9999997	9999996	9999995	...	2	1
0	1	2	3	4	5	...	9999998	9999999

Vetor C

740	911	702	641	708	349	...	952	380
0	1	2	3	4	5	...	9999998	9999999

O seu programa deve calcular o tempo de execução de cada operação nas árvores (inserção e consulta) com cada um dos vetores gerados. Você deve pegar os tempos imediatamente antes e imediatamente após a execução da operação conforme o exemplo abaixo. *Obs: caso haja estouro de pilha (stack overflow), sugira uma solução para resolver o problema.*

inicio = tempo inicial;

operação sobre a árvore

fim = tempo final;

tempo de execução = fim - inicio;

Gere uma tabela e um gráfico no Excel que compare os tempos obtidos para fazer cada uma das 4 operações sobre os 3 vetores (veja o exemplo abaixo). Observação: os tempos abaixo são aleatórios e não refletem os tempos de execução desses métodos.

	Inserção ABP	Inserção AVL	Consulta ABP	Consulta AVL
A	5	7	3	1
B	5	8	3	2
C	5	10	3	3

O que deve ser entregue: um relatório com o código fonte de todos os testes realizados; gráfico com os tempos de execução; sua análise sobre os tempos encontrados.

Pontifícia Universidade Católica de Minas Gerais

Curso: Engenharia de Software

Disciplina: Algoritmos e Estruturas de Dados II

Professor: Rodrigo Richard Gomes

Entrega: 28/05/2019

DUPLA – 20 pontos

PARTE 2

Esse trabalho consiste em implementar e testar os 6 métodos de ordenação vistos em sala de aula (BubbleSort, BubbleSort adap. 1, BubbleSort adap. 2, BubbleSort Híbrido, Inserção, Seleção e MergeSort, ShellSort e QuickSort). Você deve testar a ordenação de 3 vetores de 100 mil posições através desses métodos. Os vetores devem estar preenchidos da seguinte maneira:

- Vetor A : preenchido com os valores entre 1 e 100000
- Vetor B : preenchido com os valores entre 100000 e 1 (vetor invertido)
- Vetor C: preenchido aleatoriamente

Vetor

A

1	2	3	4	5	6	...	99999	100000
0	1	2	3	4	5	...	99998	99999

Vetor

B

100000	99999	99998	99997	99996	99995	...	2	1
0	1	2	3	4	5	...	99998	99999

Vetor

C

740	911	702	641	708	349	...	952	380
0	1	2	3	4	5	...	99998	99999

O seu programa deve calcular o tempo de execução de cada algoritmo com cada um dos vetores gerados. Você deve pegar os tempos imediatamente antes e imediatamente após a execução do método de ordenação conforme o algoritmo abaixo.

inicio = tempo inicial;

ordenação

fim = tempo final;

tempo de execução = fim - inicio;

Gere uma tabela e um gráfico no Excel que compare os tempos obtidos pelos métodos de ordenação sobre os 3 vetores (veja o exemplo abaixo). Observação: os tempos abaixo são aleatórios e não refletem os tempos de execução desses métodos.

Pontifícia Universidade Católica de Minas Gerais

Curso: Engenharia de Software

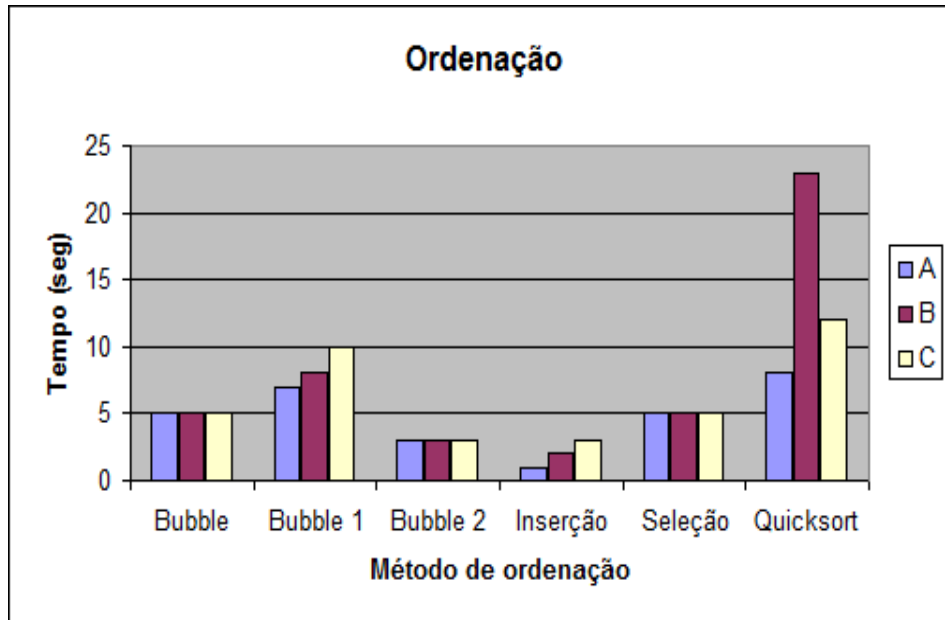
Disciplina: Algoritmos e Estruturas de Dados II

Professor: Rodrigo Richard Gomes

Entrega: 28/05/2019

DUPLA – 20 pontos

	Método de ordenação					
	Bubble	Bubble a1	Bubble a2	Bubble a3	...	Quicksort
A	5	7	3	1		8
B	5	8	3	2		23
C	5	10	3	3		12



O que deve ser entregue: um relatório com o código fonte de todos os testes realizados mais o gráfico com os tempos de execução.