UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

MELHORIAS E IMPLEMENTAÇÕES DE ALGORITMOS

Relatório para a disciplina de Estrutura de dados

MATHEUS BUENO EUZÉBIO

CORNÉLIO PROCÓPIO

2018

Melhoria em Bubble Sort

Ao utilizamos o algoritmo de ordenação Bubble sorte percebe-se que o mesmo não é eficiente. Devido a isso foi proposto que seu código fosse melhorado.

Foi utilizado uma função que verifica (a cada nova chamada do algoritmo) se o mesmo já está ordenado, caso esteja não há a necessidade de continuar o método, assim finalizando-o. O método verificaOrdenacao() realiza a verificação de ordenação da seguinte forma:

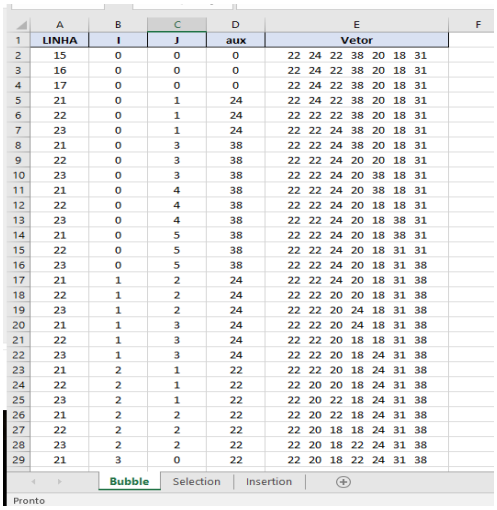
Caso o valor atual do vetor seja menor que o próximo significa que o vetor não está ordenado e retorna 0, se estiver ordenado será retornado o valor 1 e o método bubble é finalizada.

Uma outra melhoria aplicada foi a de diminuir o tamanho do vetor a cada chamada pois, a seu último valor já está na posição correta, não precisando assim passar pelo mesmo novamente, diminuindo assim a quantidade de vezes chamadas de 42 para 11.

Implementação dos três métodos

Utilizando o método Bubble sort foram necessários 32 passos para que o vetor pudesse ser organizado, já no Selection sort foram necessários 21 passos e no método Insertion foram necessários 11 passos, mostrando assim que o Insertion é o mais eficiente deles e o bubble sorte o mais ineficiente.

Através dos testes de mesa pode se perceber que que o bubble sorte é mais ineficiente que os outros dois devido ao seu método de pesquisa que mesmo com o vetor já ordenado em certas posições ele repete a operação, se tornando assim mais ineficiente.



	A	B	C	D	E	F
1	LINHA	I	J	aux	Vetor	
2	15	0	0	0	22 24 22 38 20 18 31	
3	16	0	0	0	22 24 22 38 20 18 31	
4	17	0	0	0	22 24 22 38 20 18 31	
5	21	0	1	24	22 24 22 38 20 18 31	
6	22	0	1	24	22 22 22 38 20 18 31	
7	23	0	1	24	22 22 24 38 20 18 31	
8	21	0	3	38	22 22 24 38 20 18 31	
9	22	0	3	38	22 22 24 20 20 18 31	
10	23	0	3	38	22 22 24 20 38 18 31	
11	21	0	4	38	22 22 24 20 38 18 31	
12	22	0	4	38	22 22 24 20 18 18 31	
13	23	0	4	38	22 22 24 20 18 38 31	
14	21	0	5	38	22 22 24 20 18 38 31	
15	22	0	5	38	22 22 24 20 18 31 31	
16	23	0	5	38	22 22 24 20 18 31 38	
17	21	1	2	24	22 22 24 20 18 31 38	
18	22	1	2	24	22 22 20 20 18 31 38	
19	23	1	2	24	22 22 20 24 18 31 38	
20	21	1	3	24	22 22 20 24 18 31 38	
21	22	1	3	24	22 22 20 18 18 31 38	
22	23	1	3	24	22 22 20 18 24 31 38	
23	21	2	1	22	22 22 20 18 24 31 38	
24	22	2	1	22	22 20 20 18 24 31 38	
25	23	2	1	22	22 20 22 18 24 31 38	
26	21	2	2	22	22 20 22 18 24 31 38	
27	22	2	2	22	22 20 18 18 24 31 38	
28	23	2	2	22	22 20 18 22 24 31 38	
29	21	3	0	22	22 20 18 22 24 31 38	

Bubble | Selection | Insertion

Pronto

Figura 1Teste de mesa do BubbleSort

Alem disso, percebe se também que o Insertion sort é mais eficiente devido a sua ordenação ser mais inteligente, onde se compara dois elementos e caso o primeiro for maior que o segundo o algoritmo compara com o seus anteriores até encontrar o menor, ocorrendo assim uma única passagem.

	A	B	C	D	E	F
1	**LINHA**	**I**	**J**	**n**	**aux**	**Vetor**
2	50	0	0	7	0	22 24 22 38 20 18 31
3	51	0	0	7	0	22 24 22 38 20 18 31
4	52	0	0	7	0	22 24 22 38 20 18 31
5	54	1	7	7	24	22 24 22 38 20 18 31
6	56	1	0	7	24	22 24 22 38 20 18 31
7	59	2	0	7	22	22 24 22 38 20 18 31
8	54	2	1	7	22	22 24 24 38 20 18 31
9	56	2	0	7	22	22 22 24 38 20 18 31
10	59	3	0	7	38	22 22 24 38 20 18 31
11	54	3	2	7	38	22 22 24 38 20 18 31
12	56	4	2	7	20	22 22 24 38 20 18 31
13	59	4	3	7	20	22 22 24 38 38 18 31
14	54	4	2	7	20	22 22 24 24 38 18 31
15	56	4	1	7	20	22 22 22 24 38 18 31
16	59	4	0	7	20	22 22 22 24 38 18 31
17	54	4	-1	7	20	20 22 22 24 38 18 31
18	56	5	-1	7	18	20 22 22 24 38 18 31
19	59	5	4	7	18	20 22 22 24 38 38 31
20	54	5	3	7	18	20 22 22 24 24 38 31
21	56	5	2	7	18	20 22 22 22 24 38 31
22	59	5	1	7	18	20 22 22 22 24 38 31
23	54	5	0	7	18	20 20 22 22 24 38 31
24	56	5	-1	7	18	18 20 22 22 24 38 31
25	59	6	-1	7	31	18 20 22 22 24 38 31
26	54	6	5	7	31	18 20 22 22 24 38 38
27	56	6	4	7	31	18 20 22 22 24 31 38

Figura 2Teste de mesa do InsertionSort

A planilha contém o teste de mesa dos três métodos implementados.

Onde é mostrado o valor de cada variável na linha em que há mudanças de valores de variáveis e os valores das linhas vão se repetindo porque como está dentro de um for a passagem na linha é feita várias vezes.

Para uma melhor visualização foi utilizado apenas os valores das idades no teste de mesa (o valor que é usado para as comparações), no entanto o algoritmo ordena o objeto todo e não só a idade.