

LISTA 2 | BUSCA E ORDENAÇÃO (VETORES)

Enos Andrade

- 1 Descreva e exemplifique em código comentado dois algoritmos de busca:

<http://cglab.ca/morin/misc/sortalg/>

Solução

Bubble Sort

```
1 //O bubbleSort recebe como parâmetros o tamanho e o ponteiro para o vetor
2 void bubbleSort(int tamanho, int *vetor){
3     //Variavel aux pra armazenar temporariamente um dos números a serem trocados
4     int aux;
5     //Este primeiro loop tem como objetivo verificar quantas vezes será preciso
6     //  ↳ fazer o loop interno. A quantidade vai depender do tamanho do vetor, visto
7     //  ↳ que a cada loop externo o maior número do vetor é colocado no último
8     //  ↳ índice. E por isso ele é ignorado no loop seguinte.
9     for(int fim = tamanho-1; fim > 1; fim--){
10        //Este loop interno faz as verificações e, caso necessário, as trocas
11        for(int index = 0; index < fim; index++){
12            //Verifica se o valor do vetor no index corrente(vetor[index]) é maior
13            //  ↳ que o posterior(vetor[index+1]). Caso verdadeiro fará a troca de
14            //  ↳ posição.
15            if (vetor[index] > vetor[index+1]) {
16                // aux armazenará o valor armazenado no índice corrente
17                aux = vetor[index];
18                //O valor do índice posterior será a atribuído ao corrente
19                vetor[index] = vetor[index+1];
20                // O valor da variavel aux, que armazena o valor original do
21                //  ↳ corrente, será atribuído ao posterior
22                vetor[index+1] = aux;
23            }
24        }
25    }
26 }
```

Selection Sort

```
1 void selectionSort (int tam, int *vetor){ // Recebe tamanho e vetor
2     int menor; // Pra armazenar o índice que tem o menor valor
3     int aux; //Pra auxiliar na troca
4
5     //Os valores são trocados um por um, primeiro o menor valor vai para índice 0,
6     //  ↳ depois o segundo menor valor vai para o índice 1 e assim até o final
7     for(int i = 0; i < tam-1; i++){ //Os dois ultimos valores são ordenados ao
8     //  ↳ mesmo tempo, por isso tam-1
9
10        //busca o menor elemento no array
11        menor = i;
12        for(int j = i+1; j < tam; j++){
13            if (vetor[j] < vetor[menor]) {
14                menor = j;
15            }
16        }
17    }
18 }
```

```
15 //Faz a troca, da mesma forma que acontece no Bubble Sort
16 aux = vetor[i];
17 vetor[i] = vetor[menor];
18 vetor[menor] = aux;
19 }
20 }
```

2 Escreva um programa que permita buscar um nome em um vetor de strings ordenado alfabeticamente.

– Dica: a função `strcmp(a,b)` já faz o trabalho de retornar:

- -1 se $a < b$;
- +1 se $a > b$;
- 0 se $a == b$.

Solução

```
1 #include <stdio.h>
2
3 int main(){
4     char pessoa[20];
5     char pessoas[15][20] = {"Alinne", "Dalison", "Dennys", "Edberg", "Enos",
6     ↪ "Francisco", "Felipe", "Joao", "Lucas", "Ozivan", "Pablo", "Samaronia",
7     ↪ "Severino", "Tamara", "Washington"};
8
9     printf("Buscar nome: ");
10    gets(pessoa);
11
12    for(int i = 0; i < 15; i++){
13        if (strcmp(pessoa, pessoas[i]) == 0) {
14            printf("%s esta na posicao %d", pessoa, i);
15            return 0;
16        }
17    }
18    printf("%s nao esta no vetor", pessoa);
19    return 0;
20 }
```

3 Escreva um programa que crie um vetor de ponteiros para a estrutura `Aluno` (ordenado crescentemente por nome e matricula) e permita realizar buscas por nomes nesse vetor.

```
struct aluno {
    char *nome;
    int matricula;
};
typedef struct aluno Aluno;
```

Solução

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct aluno{
5      char nome[20];
6      int matricula;
7  };
8
9  int main(){
10
11      struct aluno alunos[3];
12
13      for(int i = 0; i < 3; i++){
14          printf("Aluno %d\n", i+1);
15          printf("Nome: ");
16          setbuf(stdin, NULL);
17          gets(alunos[i].nome);
18          printf("matricula: ");
19          scanf("%d", &alunos[i].matricula);
20      }
21
22      char aluno[20];
23      printf("Buscar aluno: ");
24      setbuf(stdin, NULL);
25      gets(aluno);
26
27      for(int i = 0; i < 6; i++){
28          if (strcmp(alunos, alunos[i].nome) == 0) {
29              printf("%s esta na posicao %d", aluno, i);
30              return 0;
31          }
32      }
33      printf("%s nao esta no vetor", aluno);
34      return 0;
35 }
```