

Manipulação de Vetores

1 Programas

Neste trabalho os alunos deverão escrever vários programas, em linguagem Java ou linguagem C, que manipulam vetores. Suponha que um vetor (= *array*) v foi declarado como `int v[MAX]`, sendo MAX uma constante. Digamos que v abriga uma sequência $v[0], \dots, v[n-1]$. É óbvio que devemos ter $0 \leq n \leq MAX$. Se n é igual a 0 então a sequência está vazia. Se n é igual a MAX , a sequência está cheia. Suponha que a sequência $v[0], \dots, v[n-1]$ pode sofrer inserção de novos elementos e remoção de elementos antigos.

A descrição dos programas a serem implementados é apresentada a seguir.

Programa 1 : Implemente métodos (funções) `inserir` e `remove` um elemento de um vetor. Utilize as definições abaixo:

remove : Este método recebe $0 \leq k < n$ e remove o elemento $v[k]$ do vetor $v[0..n-1]$. O método devolve o novo tamanho do vetor. Qualquer situação que não permita a remoção, deve-se retornar -1.

inserir : Este método recebe um inteiro x e $0 \leq k < n$ e insere x no elemento $v[k]$ do vetor v . O método devolve o novo tamanho do vetor v (com a inserção de x , ou seja, com um elemento a mais). Qualquer situação que não permita a inserção, deve-se retornar -1.

Programa 2 : Refaça todo o problema da remoção sob condições mais gerais. Suponha que a parte relevante do vetor v é $v[ini..fim-1]$; para remover $v[k]$, puxe $v[k+1..fim-1]$ para a esquerda ou empurre $v[ini..k-1]$ para a direita, dependendo de qual das alternativas seja mais “barata”. O método devolve o novo tamanho da parte relevante do vetor. Qualquer situação que não permita a remoção, deve-se retornar -1.

Programa 3 : Escreva um método que insira um elemento x entre $v[k]$ e $v[k+1]$.

Programa 4 : Um segmento horizontal de um vetor $x[0..n-1]$ é um subvetor $x[p..q]$ tal que $x[p] == x[p+1] == \dots == x[q]$. O tamanho de um tal subvetor é $q-p+1$. Um segmento horizontal é *máximo* se não existe segmento horizontal de tamanho maior. Escreva um método que receba um vetor *crescente* não vazio $x[0..n-1]$ e devolva o tamanho de um segmento horizontal máximo no vetor. Procure escrever uma função simples e limpa.

Programa 5 : Decidir se $x[0..m-1]$ é subsequência de $v[0..n-1]$, ou seja, decidir se x é o que sobra depois que alguns dos elementos de v são apagados. Quer uma definição mais formal? Diz-se que $x[0..m-1]$ é subsequência de $v[0..n-1]$ se

$$x[0] = v[s[0]], x[1] = v[s[1]], \dots, x[m-1] = v[s[m-1]]$$

para algum vetor $s[0..m-1]$ tal que $0 \leq s[0] < s[1] < \dots < s[m-1] \leq n-1$. Por exemplo, 12,13,10,3 é uma subsequência de 11,12,13,11,10,9,7,3,3 mas não de 11,12,10,11,13,9,7,3,3. Vocês devem escrever um programa que diga se um vetor x é subsequência de um vetor $v[0..n-1]$.

Programa 6 : Escreva um (a) método (função) que remova de um vetor $v[ini..fim-1]$ todas as ocorrências de um elemento y .

Programa 7 Escreva um método recursivo `onde()`. Ao receber um inteiro x , um vetor v e um inteiro n , o método deve devolver j no intervalo fechado $0..n-1$ tal que $v[j]$ é a primeira ocorrência em que $v[i] == x$; se tal j não existe, o método deve devolver -1 .

Programa 8 Escreva um método recursivo que recebe um inteiro x , um vetor v e inteiros `ini` e `fim` e devolve j tal que $ini \leq j \leq fim-1$ e $v[j]$ é a primeira ocorrência em que $v[j]$ é igual a x ; se tal j não existe então devolve -1 .

Programa 9 : Modifique o algoritmo para determinar o valor da subsequência máxima (*GlobalMax*) (veja slides sobre *Indução Forte*) para retornar um vetor que contenha uma possível subsequência máxima.

Todos programas valem 1,0 ponto; exceto o Programa 9, que vale 2,0 pontos.

2 Descrição dos parâmetros e valores de retorno

Todos os programas deverão ser implementados na forma de métodos (para Java) ou funções (para C). A classe Java deverá se chamar **Vetor** e deverá conter um método para cada programa descrito acima. Para quem fizer em linguagem C, deverá entregar um arquivo chamado com uma função implementando cada um dos programas.

A seguir são descritas as assinaturas dos métodos (Java) e os protótipos das funções (C) para cada programa. Observem que o valor de retorno e os parâmetros deverão ser seguidos rigorosamente pois a correção será automática por meio da invocação dos métodos ou funções.

Programa 1a : k é a posição do vetor cujo elemento será removido; n é o tamanho do vetor v . O valor de retorno é o novo tamanho do vetor.

Java : `int remover1 (int k, int n, int v[]);`

C : `int remover1 (int k, int n, int * v);`

Programa 1b : x é o valor a ser inserido; k é a posição do vetor na qual o novo elemento será inserido; n é o tamanho do vetor v . O valor de retorno é o novo tamanho do vetor.

Java : `int inserir1 (int x, int k, int n, int v[]);`

C : `int inserir1 (int x, int k, int n, int * v);`

Programa 2 : n é o tamanho do vetor v . `ini` e `fim` são as posições de início e fim, respectivamente, da parte relevante do vetor v . $ini \leq k < fim$ é a posição do elemento a ser removido.

O valor de retorno é o novo tamanho da parte relevante do vetor.

Java : `int remover2 (int k, int n, int ini, int fim, int v[]);`

C : `int remover2 (int k, int n, int ini, int fim, int * v);`

Programa 3 : o novo elemento x deve ser inserido depois da posição k do vetor v ; n é o tamanho do vetor v .

Java : `void inserir3 (int x, int k, int n, int v[]);`

C : `void inserir3 (int x, int k, int n, int * v);`

Programa 4 : n é o tamanho do vetor v ; O valor de retorno é o tamanho do segmento horizontal máximo.

```
Java : int tamanhoSegmentoHorizontal (int n, int v[]);  
C : int tamanhoSegmentoHorizontal (int n, int * v);
```

Programa 5 : tamX é o tamanho do vetor x ; n é o tamanho do vetor v . O valor de retorno é *true* (1, para C) se o vetor x é subsequência do vetor v ; e *false* (0, para C), caso contrário.

```
Java : boolean ehSubsequencia (int tamX, int x[], int n, int v[]);  
C : int ehSubsequencia (int tamX, int * x, int n, int * v); 1 -- true; 0 -- false
```

Programa 6 : y é o valor a ser removido; n é o tamanho do vetor v . ini e fim são as posições de início e fim, respectivamente, que indicam a parte relevante do vetor v , isto é, $v[\text{ini}] \dots v[\text{fim}-1]$.

```
Java : void remover6 (int y, int n, int ini, int fim, int v[]);  
C : void remover6 (int y, int n, int ini, int fim, int * v);
```

Programa 7 : x é o valor a ser localizado; n é o tamanho do vetor v . O valor de retorno é a posição j onde x está localizado; ou -1, caso não esteja presente no vetor v .

```
Java : int onde (int x, int n, int v[]);  
C : int onde (int x, int n, int * v);
```

Programa 8 : x é o valor a ser localizado; n é o tamanho do vetor v . ini e fim são as posições de início e fim, respectivamente, que indicam a parte relevante do vetor v , isto é, $v[\text{ini}] \dots v[\text{fim}-1]$. O valor de retorno é a posição $\text{ini} \leq j \leq \text{fim}-1$ onde x está localizado; ou -1, caso não esteja presente no vetor v .

```
Java : int localiza (int x, int n, int ini, int fim, int v[]);  
C : int localiza (int x, int n, int ini, int fim, int * v);
```

Programa 9 : n é o tamanho do vetor v . O valor de retorno é um ponteiro para um vetor que contém a subsequência máxima. Para C, há um parâmetro adicional (**tamret**) que é o tamanho do vetor retornado. Esse parâmetro é passado por referência e deverá ser atualizado dentro da função `subSequenciaMaxima`.

```
Java : int [] subSequenciaMaxima (int n, int v[]);  
C : int * subSequenciaMaxima (int n, int * v, int * tamret);
```

3 Entrega dos programas

- O EP é individual. Trabalhos copiados serão punidos com nota *Zero* tanto para o *plagiador* como para o *plagiado*.
- Os alunos que desenvolverem em Java deverão entregar a implementação da classe **Vetor** em um arquivo renomeado como **NumeroUSP.java**. Aqueles que implementarem em C deverão entregar um arquivo chamado **NumeroUSP.c**. Esses arquivos deverão conter apenas os métodos ou funções descritas na Seção 2 com os respectivos parâmetros. **Não** deverão conter método ou função do tipo **main**.
- A data máxima de entrega é 25 de setembro de 2019 e a forma de entrega é utilizando o Tidia.