

Trabalho 1

Apresentação: 26 e 28/09

Contagem de operações e tempo de execução

Você deverá implementar 4 programas diferentes, contabilizar o tempo médio de execução para diferentes entradas, construir um gráfico a partir disso, e contabilizar suas operações.

Os programas a serem implementados são:

1. Inverter a ordem do vetor de entrada
2. Busca Sequencial no vetor de entrada
3. Busca Binária Iterativa no vetor de entrada
4. Busca Binária Recursiva no vetor de entrada

Seus formatos de entrada e saída estão especificados abaixo, enunciados disponíveis também no run.codes.

A avaliação consistirá de 4 partes, cada uma valendo 25% da nota final:

1 Implementação e corretude

Os programas devem ser implementados e submetidos no site <https://runcodes.icmc.usp.br/>, acessando a turma com o código de disciplina GR9C. A nota final dessa parte será dada pela média da nota obtida nos casos de teste dos exercícios.

OBS.: Para o exercício "Buscas no Grande Vetor", devem ser submetidas DUAS implementações, uma iterativa e outra recursiva.

2 Organização e boas práticas

Após o período de submissão os códigos serão revisados buscando boas práticas no desenvolvimento de programas.

Recomendamos o material "Guia de Estilos para ICC I", disponível na página do Tidia.

3 Relatório

Você deverá fazer um relatório simples, contendo a resolução da equação de contagem de operações dos algoritmos, no PIOR CASO, e o gráfico resultante das execuções.

Utilize, pelo menos, os seguintes tamanhos de entrada: 10 , 10^2 , 10^3 e 10^4 .

Faça a média de tempo de, pelo menos, 10 execuções.

Compare os 4 algoritmos e comente os resultados. O que os gráficos indicam? Qual a complexidade de tempo no pior caso?

4 Seminário

Deverá ser apresentado um seminário de 15 minutos nos dias 26 e 28/09, explicando as implementações realizadas e os resultados obtidos.

Referências

Pereira, Leonardo, "*Listas de Exercícios Estruturas de Dados I*", ICMC-USP, 2022.

Inverter Vetor

Dado um vetor com N elementos, inverta a sequência desses elementos. Isto é, dado um vetor com elementos x_1, x_2, \dots, x_n , obtenha outro vetor com elementos x_n, x_{n-1}, \dots, x_1 .

Entrada

A primeira linha contém um inteiro N , o número de elementos. Na linha abaixo, estão N inteiros, os valores do vetor.

Saída

A saída consiste de uma linha, contendo os valores do vetor na ordem invertida.

Restrições

- ($1 \leq N \leq 10^5$)

Exemplo 1

Entrada	Saída
10 1 2 3 4 5 6 7 8 9 10	10 9 8 7 6 5 4 3 2 1

Exemplo 2

Entrada	Saída
7 -1 6 3 -5 7 7 2	2 7 7 -5 3 6 -1

Buscas no Vetor

Dado um vetor com N elementos, realize Q consultas, onde cada consulta consiste em ler um número e dizer se ele está presente ou não no vetor.

Entrada

A primeira linha contém um inteiro N , o número de elementos. Na linha abaixo, estão N inteiros, os valores do vetor. A terceira linha contém o inteiro Q , o número de consultas a serem realizadas. Em seguida seguem Q linhas, cada uma contém um inteiro representando o número a ser buscado naquela pesquisa.

Saída

Para cada consulta, deve ser imprimido "SIM" caso o inteiro buscado estiver no vetor e "NAO" caso contrário.

Restrições

- $(1 \leq N \leq 10^3)$
- $(1 \leq Q \leq 10^3)$

Exemplo 1

Entrada	Saída
5	NAO
3 7 10 13 17	NAO
5	SIM
1	SIM
2	NAO
3	
10	
15	

Exemplo 2

Entrada	Saída
10	SIM
4 9 17 -1 -6 0 20 4 5 7	NAO
3	SIM
4	
1	
7	

Buscas no Grande Vetor

Dado um vetor com N elementos ordenados únicos, realize Q consultas nesse vetor, onde cada consulta consiste em ler um número e buscar por sua posição no vetor.

Entrada

A primeira linha contém um inteiro N , o número de elementos. Na linha abaixo, estão N inteiros, os valores do vetor. A terceira linha contém o inteiro Q , o número de consultas a serem realizadas. Em seguida seguem Q linhas, cada uma contém um inteiro representando o número a ser buscado naquela pesquisa.

Saída

Para cada consulta, deve ser imprimida uma linha, contendo a posição do elemento no vetor caso ele seja encontrado, ou "-1" caso contrário.

Restrições

- $(1 \leq N \leq 10^5)$
- $(1 \leq Q \leq 10^5)$

Exemplo 1

Entrada	Saída
10	1
1 2 8 11 18 20 21 35 42 57	-1
5	8
2	-1
41	5
42	
19	
20	

Exemplo 2

Entrada	Saída
5	4
-7 -3 0 4 9	-1
3	2
9	
-1	
0	

OBS.: Devem ser submetidas DUAS soluções para esse problema, uma iterativa e outra recursiva.