

AED2 2022 (1s) - EXERCÍCIO 5

Instruções:

1. E/S: tanto a entrada quanto a saída de dados devem ser "secas", ou seja, não devem apresentar frases explicativas. Siga o modelo fornecido e apenas complete as partes informadas (veja o exemplo abaixo).
2. Identificadores de variáveis: escolha nomes apropriados
3. Documentação: inclua cabeçalho, comentários e indentação no programa.
4. Submeta o programa no sistema judge: <http://judge.unifesp.br/aed2S01A2022/>

Descrição:

A prefeitura da cidade de Quadradônia estava passando por uma grande dificuldade: muitas pessoas precisavam se vacinar, porém não havia doses para todos. O Prefeito – um político espertalhão – não queria ser culpado pela falta de vacinas. Então, pensou em acalmar os ânimos da população realizando um sorteio. Por isso, pediu a Tony Hoare¹, seu acessor e matemático, que pensasse em algo para lhe ajudar.

Tony então propôs um sorteio composto de duas etapas:

*Na primeira etapa, um número-chave aleatório era sorteado sem repetição e distribuído a cada um dos habitantes. Na segunda etapa, o k -ésimo número-chave (em ordem crescente) era chamado para tomar a vacina, **sem a necessidade de ordenar todos os números-chave**.*

Para que não se levantassem suspeitas, ninguém sabia qual era a ordem dos números-chave. Por isso, Tony convocava os cidadãos usando apenas a posição k de cada número-chave da sequência. Por exemplo, se quisesse chamar o cidadão que ocupava a posição 3 na sequência, aplicava seu algoritmo e então conhecia-se o número-chave na posição 3. Dessa forma, a sequência dos números-chave continuava oculta. Além disso, como a progressão numérica dos números-chave era desconhecida, quem fosse chamado não saberia sua posição na fila apenas pelo seu número-chave.

Escreva um programa que implemente o sorteio proposto por Tony Hoare. Seu programa **deve** utilizar o algoritmo de ordenação *QuickSort*.

Considere as seguintes condições:

1. Sua solução deve implementar a versão **recursiva** do *QuickSort*;
2. A complexidade do algoritmo deve se manter em $O(n \log n)$;
3. O pivô inicial de ser igual ao último elemento;
4. O código-fonte **deve** ser escrito em C/C++ ou Java;
5. **Toda** memória alocada dinamicamente (C/C++) deve ser desalocada;

¹https://pt.wikipedia.org/wiki/Charles_Antony_Richard_Hoare

6. Nenhuma variável global deve ser utilizada;

Entrada:

A primeira linha contém o valor de um inteiro $k | k \leq N$ que representa a posição do número-chave que queremos descobrir.

A segunda linha contém o valor de um inteiro positivo N , que representa a quantidade de números-chave.

A terceira linha contém uma lista não-ordenada de inteiros distintos separados por um espaço em branco, representando a lista de números-chave atribuídos a cada cidadão na primeira etapa do sorteio.

Saída:

A primeira linha contém um texto que apresenta quem é o elemento na posição de interesse k . Veja os exemplos abaixo.

A segunda linha, por sua vez, apresenta os números-chave da lista de entrada dispostos da seguinte forma: se x_k é o número-chave de interesse na posição k , então todos os números-chave $x \leq x_k$ devem anteceder x_k , e todos os números-chave $x > x_k$ devem suceder x_k . Tanto predecessores quanto sucessores de x_k **não** estarão necessariamente ordenados.

Exemplos de entrada e saída:

Note que a posição impressa na saída é uma a mais que a posição do elemento no vetor, cuja primeira posição é a de índice zero e não um.

Exemplos de entrada	Exemplos de saída
3 6 1 3 5 7 4 2	3o menor elemento eh o 3 1 2 3 7 4 5
1 6 2 4 3 7 15 1	1o menor elemento eh o 1 1 4 3 7 15 2
2 10 4 5 8 1 20 6 11 19 14 13	2o menor elemento eh o 4 1 4 5 6 8 11 13 19 14 20

Tabela 1: Exemplos de entrada e saída