

Trabalho 01 - Problema da k-Seleção

Data de entrega: 31/10/2020

Importante:

- **Não** olhe códigos de outros grupos ou da internet. Exceto o que é fornecido.
- O trabalho é individual.
- Em caso de plágio, fraude ou tentativa de burlar o sistema será aplicado nota 0 na disciplina aos envolvidos.
- Alguns alunos podem ser solicitados para explicar com detalhes a implementação.
- Passar em todos os testes do run.codes não é garantia de tirar a nota máxima. Sua nota ainda depende do cumprimento das especificações do trabalho, qualidade do código, clareza dos comentários, boas práticas de programação e entendimento da matéria demonstrada em possível reunião.
- Você deverá submeter, até a data de entrega, o seu código na plataforma run.codes. Seu cadastro deve ter o nome e número de matrícula exatamente como está no Sigaa.

Neste trabalho você deverá implementar um Algoritmo para o Problema da k-Seleção, onde é dado um arranjo A com n números inteiros e um arranjo ordenado i com k números entre 1 e n . O objetivo é encontrar o $i[0]$ -ésimo menor elemento de A , o $i[1]$ -ésimo menor elemento de A , ..., até o $i[k-1]$ -ésimo menor elemento de A .

Por exemplo: Se $A = \{10, 2, 3, 100, 6\}$ e $i = \{1, 2, 5\}$, a solução para essa instância é composta pelo primeiro menor valor, o segundo menor valor e quinto menor valor, portanto, os valores 2, 3 e 100.

Neste trabalho o seu programa deverá ler dois inteiros, que representam a quantidade n de valores que serão gerados aleatoriamente para formar A , e a quantidade k de estatísticas de ordem que serão geradas para formar i . Um código que já faz isso será fornecido, sugiro não mudar essa parte do programa pois é preciso garantir que os números gerados sejam sempre os mesmos.

Veja um exemplo de entrada:

10 4

O seu programa então vai gerar 10 valores aleatórios e 4 estatísticas de ordem. Nesse caso o arranjo A gerado será (1646153577, 2013625315, 66753433, 92330671, 1393706467, 1677202872, 1957184147, 1249891912 401185319, 1885161542) e o vetor i gerado será (2, 2, 5, 6) portanto seu programa deverá imprimir, o segundo menor elemento (duas vezes) depois o quinto e o sexto menor elemento:

92330671 92330671 1393706467 1646153577

Você receberá um arquivo *kSelect.c* que contém uma implementação que ordena o vetor *A* com o QuickSort e imprime $A[i[j]]-1$ para cada *j* entre 0 e *k*-1. Infelizmente apesar do QuickSort ser muito eficiente, ele não consegue resolver os testes em menos de 1 segundo, o que o seu algoritmo deve fazer. Você deve manter a forma como ele sorteia os valores.

- Você deverá implementar em linguagem c, isso porque é preciso que o gerador de números aleatórios seja sempre igual.
- Seu programa deve executar no run.codes em menos de 1 segundo.
- Você não deve usar nenhuma função pronta muito complexa.
- Se você não tiver certeza se alguma coisa é permitida ou não no trabalho, não hesite em perguntar ao professor!
- Não deixe para os últimos dias!