Trabalho 03 - Algoritmo de Seleção Aleatorizado

Data de entrega: 16/10/2020

Importante:

- **Não** olhe códigos de outros grupos ou da internet. Exceto o que é fornecido.
- TODOS os membros do grupo devem participar e compreender completamente a implementação.
- 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, onde o "Número de Matrícula" deverá ser o número do seu grupo.

Esse trabalho deverá ser realizado em grupo, com os grupos já definidos na disciplina. Neste trabalho o seu grupo deverá implementar o Algoritmo de Seleção Aleatorizado visto em aula para o Problema da Seleção, que utilizaremos para encontrar a mediana (mas deve ser geral). (Sugestão: normalmente algoritmos de divisão e conquista são implementados de maneira recursiva, o que facilita bastante o trabalho).

Neste trabalho o seu programa deverá simplesmente ler um inteiro, que representa a quantidade de valores que serão gerados aleatoriamente, 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
```

O seu programa então vai gerar 10 valores aleatórios e retornar a mediana, ou seja o 5º menor valor que foi gerado. Nesse caso os valores gerados serão (1646153577, 2013625315, 66753433, 92330671, 1393706467, 1677202872, 1957184147, 1249891912 401185319, 1885161542) e portanto seu programa deverá imprimir:

```
1393706467
```

Você receberá um arquivo *selecao.c* que contém uma implementação que ordena o vetor com o QuickSort e devolve a mediana. 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 pode deve usar nenhuma função pronta muito complexa.
- Você PRECISA implementar a versão aleatorizada. Como os valores foram sorteados aleatoriamente, é possível que uma escolha mais trivial, como escolher o primeiro elemento, passe em todos os casos de teste. Mas não passaria se os valores estivessem ordenados por exemplo.
- Se você não tiver certeza se alguma coisa é permitida ou não no trabalho, não hesite em perguntar ao professor!