

# Trabalho I - Estrutura de Dados III

## Quem é o Impostor?

Prof. Élder F. F. Bernardi ([elder.bernardi@passofundo.ifsul.edu.br](mailto:elder.bernardi@passofundo.ifsul.edu.br))

Estrutura de dados III - Ciência da Computação

IFSul - Câmpus Passo Fundo

### 1. O problema

A Agência Espacial da Sbórnia precisa de sua ajuda. Houve um grande acidente na Nave da nação enquanto estava em missão no Planeta X9. Suspeita-se de uma sabotagem alienígena. A tripulação de dez astronautas acabou sendo evaporada e a nave acabou caindo em solo hostil, no planeta invadido. A Nação Sborniana está em luto e quer descobrir quem foi o causador de tal tragédia.

Logo após a tragédia, a base da Agência Espacial conseguiu decifrar a enigmática e aflita última mensagem vinda da comunicação da nave, que dizia: “***There is 1 impostor among us***”. A mensagem provocou uma investigação dos fatos e levantando suspeitas sobre uma conspiração alienígena para sabotagem.

Diante do fato, agentes secretos infiltrados no Planeta X9 iniciaram seus trabalhos. Os agentes fizeram uma descoberta muito importante: graças a meios não divulgados, descobriram que o inimigo possui o log de tarefas da nave, na ordem correta em que aconteceram os eventos e que o usuário que realizou uma tarefa na ordem X, da lista ordenada de tarefas, é o impostor. Ao saber disto, o inimigo destruiu sua cópia da lista de tarefas, negando qualquer acusação, restando somente os logs originais, que estão nave.

Todas as tentativas de resgatar os logs do computador central da nave falharam. Então a ideia que o serviço secreto teve foi: acoplar um programa ao computador local e realizar a ordenação dos logs em loco, e então descobrir o impostor. O problema é que o serviço secreto não manja muito de algoritmos e tentou realizar esta ordenação com o famigerado *Selection Sort*. O que aconteceu é que o tempo que o algoritmo demorou para executar foi o suficiente para que as hostes inimigas descobrissem a missão, fazendo com que os agentes tivessem que desistir antes de conseguirem descobrir o Impostor.

De acordo com as descobertas, são 2 milhões de registros de log desordenados na nave e a única forma de conseguir ordená-los em tempo hábil é com um algoritmo de ordenação que seja capaz de ordenar em **tempo linear**. Também se sabe que o computador que restou na nave é péssimo em realizar comparações de grandeza ( $<$ ,  $>$ ) e por isso **este tipo de operação não poderá ser usada para realizar a ordenação**.

Sua missão, portanto, é projetar o algoritmo de ordenação, apresentando um embasamento científico para que sua solução convença o comitê de segurança. Use o algoritmo de ordenação deles como uma base para comparar o seu, mas acima de tudo procure chegar em uma solução de complexidade de tempo linear, embasando a sua prova o melhor possível.

Seu programa deverá ser capaz de mostrar quem é o impostor quando for dada a entrada da linha de log suspeita. A ordenação deve ser feita em tempo real, na nave.

Apure! Pois as relações diplomáticas entre a Sbórnia e o Planeta X9 estão se deteriorando e uma guerra parece iminente!

## 2 . Informações sobre os dados conhecidos dos arquivo de log

- Formato dos dados: JSON
- Número de entradas: 2 milhões
- Campos de cada entrada:
  - month: String
    - mês que a entrada de log foi gerada
  - log: Integer
    - número do log naquele mês
  - msg: String
    - descrição da ordem da tarefa
  - user: String
    - usuário do sistema que realizou a tarefa
- O fator de ordenação é o primeiramente o mês e depois o número do log no mês.
- O fator de ordenação é primeiramente o mês e depois o número do log no mês. Alguns meses podem ter dois ou mais números de log idênticos; neste caso, a ordenação dos registros deve se dar na ordem em que aparecem originalmente, pois se tratam da mesma tarefa sendo feita por dois usuários. Ou seja, **o algoritmo deve respeitar a estabilidade dos dados de entrada**.
- O arquivo de entrada está disponibilizado no link do Moodle. Este arquivo é o arquivo que estará disponível na nave.
- Para fins de teste, pode-se diminuir a quantidade de entrada para medir os tempos do algoritmo com tamanhos de entradas diferentes.
- Dica: dificilmente o algoritmo terá um bom desempenho se não for programado com acesso direto à memória.

## 3. O artigo

Os elementos desejados no artigo são:

- Introdução e contextualização do cenário;
- Caracterização do problema;
- Introdução técnica para compreensão do problema;
- Apresentação da(s) soluções propostas, contendo:
  - Descrição da ideia geral da solução e técnicas utilizadas;
  - Construção e explicação do(s) algoritmo(s) da solução proposta;
- Testes de desempenho da solução
  - Descrição de como serão implementados o ambiente de teste:
    - Implementação da solução;
    - metodologia de testes;

- Tempo de execução de cada solução para os casos:
  - Melhor, pior e médio;
  - Para entradas razoavelmente pequenas, médias e grandes.
  - Ter um parâmetro de desempenho em relação ao Selection Sort;
- Otimizações e/ou mudanças realizadas
  - A cada proposta de solução e seus testes, pode-se propor modificações na solução a fim de aprimorá-la.
  - Verificar como a solução se comporta com grandes quantidades de números, qual o limite da solução;
  - Apresentar soluções hipotéticas para melhorar este limite;
  - **Leve em conta o tipo de dado que está sendo ordenado;**
- Finalização e considerações sobre o problema e a solução proposta.
- **Indicação de prováveis impostores, dado algumas entradas.**

Os elementos acima não precisam ser organizados exatamente nesta ordem, mas devem estar presentes. Espera-se artigos de 4 a 8 páginas, dentro do formato proposto.

  - Escrita formal no formato de artigos da SBC ou ABNT.

## 4. Cronograma de entregas do trabalho

02.04 - Organização inicial do trabalho;

09.04 - Primeira entrega: plano de solução do problema, implementações iniciais;

16.04 - Escrita do artigo - entrega parcial para avaliação;

23.04 - Apresentações dos trabalhos;