



Projeto de Estruturas de Dados II.
Prof. Saulo Queiroz

1. Sua equipe de até quatro integrantes receberá (possivelmente via Moodle) uma massa de dados de $\approx 4\text{GB}$ consistindo de alguns milhões de registros de dados de uma aplicação qualquer. Cada registro tem 1024 Bytes e obedece a especificação do seguinte tipo abstrato de dados:

```
struct registro
{
    unsigned long chave; // Campo chave Valor não se repete.
    unsigned long naochave; // Campo qualquer. Valor pode repetir.
    unsigned char outros[1008]; // outros campos: 1008 bytes (dados)
};
```

Qual a missão?

Resposta curta: Fazer um programa que imprima na tela o(s) registro(s) correspondentes a um dado valor de entrada correspondente ao campo `naochave`. Note que como o valor dado não é chave pode haver mais de um registro. **Todos os registros encontrados devem ser impressos na tela.**

O que entregar?

(a) FONTES EM C

- i. programa em C compilável em ambiente Linux com gcc-5 (ou equivalente) que permita ao professor informar o valor `naochave` de entrada e escolher uma DENTRE DUAS ESTRATÉGIAS DE BUSCA DISTINTAS implementadas pela equipe (uma das estratégias deve, essencialmente, basear-se em estrutura de dados não estudada em sala). Como resposta o programa deverá imprimir na tela o(s) registro(s) e uma estimativa do tempo necessário para completar a busca.
- ii. bibliotecas E arquivos de cabeçalhos correspondentes à(s) estrutura(s) de dados utilizada(s) para resolver o problema. **TODO O PROJETO DEVE SER MODULARIZADO EM BIBLIOTECAS.**

(b) Breve Relatório de projeto contendo:

- i. Aspectos teóricos contendo ao menos: breve menção e explicação de ao menos uma estratégia que poderia resolver o problema mas que é inadequada no quesito eficiência. Explicar por que são inadequadas; Descrição das duas estratégias de busca adotadas. Acompanhada da descrição deve vir um breve texto justificando a escolha da estrutura. **AMBAS AS ESTRATÉGIAS DA EQUIPE DEVEM PRIORIZAR A REDUÇÃO DO TEMPO DE BUSCA E, EM SEGUNDO LUGAR, A ECONOMIA DE ESPAÇO.**
- ii. Aspectos práticos contendo ao menos: breve explicação de como a estratégia foi implementada: lista de bibliotecas, arquivos e protótipos das funções mais importantes (explicar o que faz: entrada, processamento e saída); para cada função indicar seu responsável (somente 1 aluno pode ser o responsável independentemente de mais de um colaborar na mesma função);

- A. ajuste/definição de parâmetros [QUANDO PERTINENTE]. Se houver algum parâmetro nas estruturas escolhidas (muito comum em *hash*), explicar como foram definidos. Informar caso tenham feito testes preliminares para tomada de decisão.
- B. instruções de compilação do programa e das bibliotecas.
- C. explicação da metodologia para medição dos tempos de busca pedidos.
- D. projetar um ou mais testes comparativos para se ter ideia sobre qual das estratégias é mais eficiente. Ex.: *Você pode estimar a melhor estrutura como sendo aquela que apresenta melhor desempenho para um determinado sub-conjunto de registros que caiba na memória RAM. Relembre suas aulas de estatística para determinar o tamanho desse conjunto. Não esqueça de descrever brevemente a metodologia adotada no relatório.*
- E. outros aspectos que se manifestarem relevantes para o projeto (especialmente no quesito de desempenho) e que não tenham sido mencionados anteriormente.

Suposições e recomendações gerais:

1. BLOCO DE DISCO: 4096 bytes.
2. A memória RAM do computador do prof. não tem 4 GB sobrando para carregar o arquivo inteiro. *Além disso, só será tolerada a varredura sequencial do arquivo para a construção do índice em RAM.*
3. *Os trabalhos serão corrigidos em ambiente Linux com gcc-5. Erros de compilação ou mal-funcionamento anularão a nota do projeto.*