



Busca e Ordenação

Escreva um programa para ler um arquivo contendo os dados de 30228 escolas no ENEM. Cada linha do arquivo representa uma escola e é composta por **nove** informações, separadas por ponto e vírgulas (use a função **strtok**):

- id
- estado
- municipio
- rede
- media_ciencias_natureza
- media_ciencias_humanas
- media_linguagem
- media_matematica
- media_redacao

```
1 4810;MG;ABADIA DOS DOURADOS;State;458.4723404;462.2851064;504.0893617;527.9489362;551.0638298
2 7608;GO;ABADIANIA;Private;502.2944444;526.8194444;546.9583333;586.4138889;495
3 2211;MG;ABAETE;Private;564.7375;564.275;598.884375;680.746875;670
4 4474;MG;ABAETE;State;495.015534;492.9485437;537.0631068;590.8805825;548.3495146
5 2515;PA;ABAETETUBA;Private;507.4733333;529.6066667;532.96;520.9266667;585.3333333
6 3983;PA;ABAETETUBA;State;405.8777778;402.0111111;444.0074074;414.3925926;401.4814815
7 4033;PA;ABAETETUBA;Private;525.6683333;525.8566667;552.955;553.4783333;690
8 5902;PA;ABAETETUBA;State;468.298062;477.9732558;510.7573643;489.2112403;550.9302326
```

Os dados de cada escola devem ser armazenados em um array de estruturas. Em seguida, implemente os algoritmos de **busca binária** e **sequencial (sem ordenar)** para a operação de busca pelo ID da escola.

Sabendo que a ordenação somente deve ser utilizada quando a busca é uma tarefa frequente, faça uma estatística do tempo que as funções implementadas consomem a medida que se aumenta o número de operações de busca (por exemplo, 1000, 2000, 3000 operações de busca). Meça o tempo de cada conjunto de buscas utilizando a função **clock()** (veja um exemplo de uso da função mais abaixo).

Compare o tempo da busca com o tempo para realizar a ordenação dos dados (use a função **qsort**). De modo a garantir uma comparação justa, utilize o arquivo **indices.txt** como entrada para as operações de busca. Este arquivo contém um total de 1.000.000 de valores de ID's.

É possível identificar uma quantidade mínima de buscas de modo que compense o custo de ordenar os dados?

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main() {
    clock_t inicio, fim;
    unsigned long int tempo;
    inicio = clock();
    //=====
    /* coloque seu código aqui */
    //=====
    fim = clock();
    tempo = (fim - inicio)*1000/CLOCKS_PER_SEC;
    printf("tempo: %lu milissegundo\n", tempo);

    return 0;
}
```