IMD0029 – ESTRUTURAS DE DADOS BÁSICAS I

PROF. EIJI ADACHI M. BARBOSA

Roteiro de Implementação – Algoritmos de Busca

Nesta atividade prática de implementação, iremos implementar os algoritmos de busca sequencial e busca binária em suas versões iterativas e recursivas, como visto em sala de aula. Lembre-se que o problema da busca pode ser definido como o problema de, dada uma chave K e um array V, encontrar um índice X tal que V[X] == K.

Para auxiliar esta atividade, disponibilizei um conjunto de testes executáveis para que você verifique a sua implementação. Junto a este roteiro, você encontrará os seguintes arquivos:

- main.cpp – Implementa os testes executáveis.
- Search.hpp – Define a interface do módulo de busca.
- IteLinSearch.cpp – Implementa o módulo de busca usando o algoritmo de busca linear iterativo. Já está implementado.
- RecLinSearch.cpp – Implementa o módulo de busca usando o algoritmo de busca linear recursivo. Não está implementado; você deve implementá-lo.
- IteBinSearch.cpp – Implementa o módulo de busca usando o algoritmo de busca binária iterativo. Não está implementado; você deve implementá-lo.
- RecBinSearch.cpp – Implementa o módulo de busca usando o algoritmo de busca binária recursivo. Não está implementado; você deve implementá-lo.

Além destes arquivos, você precisa baixar o arquivo input.txt, o qual contém os dados de entrada nos quais nossas funções realizarão suas buscas. Baixe este arquivo em:
https://drive.google.com/file/d/0B2wvNH3OO2iVc0RmSXJwT1hteGc/view?usp=sharing

Para iniciar esta atividade, abra o arquivo main.cpp e leia com cuidado os testes implementados. Veja que na função *testSearch* há um *for* que chama várias vezes a função search, verificando justamente se a condição que define o problema de busca, como mostrada acima, é obedecida. Em seguida, há uma série de outros testes verificando que a sua função de busca é capaz de identificar os casos em que se busca uma chave que não está contida no array de entrada.

Em seguida, compile os arquivos IteLinSearch.cpp e main.cpp criando um executável chamado ILS (ILS = Iterative Linear Search):

```
g++ IteLinSearch.cpp main.cpp -o ILS -Wall -pedantic -std=c++11
```

E execute o programa gerado (certifique-se de que o arquivo input.txt está no mesmo diretório do arquivo executável):

```
./ILS
```

Você deverá ver a seguinte mensagem:

```
All tests passed!
```

Agora, você deverá implementar a função de busca nos outros módulos. Para isso, você deverá implementar a função de busca dentro de cada um destes arquivos: RecLinSearch.cpp, IteBinSearch.cpp e RecBinSearch.cpp. Para cada módulo implementado, compile e crie um executável diferente:

```
g++ RecLinSearch.cpp main.cpp -o RLS -Wall -pedantic -std=c++11
g++ IteBinSearch.cpp main.cpp -o IBS -Wall -pedantic -std=c++11
g++ RecBinSearch.cpp main.cpp -o RBS -Wall -pedantic -std=c++11
```

RLS significa *Recursive Linear Search*, **IBS** significa *Iterative Binary Search* e **RBS** significa *Recursive Binary Search*. Uma vez que você tenha terminado de implementar as diferentes funções de busca, compare o tempo de execução de cada programa. Para isto, use o comando *time* da seguinte forma:

```
time ./ILS
```

Você deverá ver uma mensagem da seguinte forma:

```
All tests passed!
real  0m3.734s
user  0m3.700s
sys   0m0.017s
```

Qual foi o programa mais rápido? E qual foi o mais lento?