Universidade Federal do Rio Grande do Norte - UFRN
Instituto Metrópole Digital - IMD
Bacharelado em Tecnologia da Informação - BTI

Disciplina: Linguagem de Programação I - LP1
Docentes: Everton Cavalcante
Silvio Sampaio
Discente: Joaliton Luan

Laboratório 9 - Namespace e bibliotecas

1. Introdução

Este relatório tem como objetivo explicar de forma sucinta a metodologia utilizada para implementação e desenvolvimento do Laboratório 9. Para tanto, criou-se uma biblioteca com implementações genéricas de algoritmos de busca, ordenação, pilha, fila, *deque*, lista vistos em Estrutura de Dados Básicos I (EDB1).

2. Metodologia

A metodologia desenvolvida baseou-se nos conhecimentos vistos em sala de aula na matéria de Linguagem de Programação 1, no que diz respeito ao sistema de versionamento (*Git/GitHub*), modularização e documentação automática no padrão *Doxygen*. Foi-se implementado uma biblioteca, nas formas dinâmicas e estática, para *Linux* e *Windows*, contendo 6 algoritmos de busca, 6 algoritmos de ordenação e 4 tipos abstratos de dados, dos quais podemos descrever:

2.1 Buscas

Dos algoritmos de busca, foram implementados a busca sequencial, a busca binária e a busca ternária, nas suas formas recursivas e iterativas, totalizando os 6 algoritmos implementados. A assinatura das funções pode ser vista à seguir:

*** T buscaSequencialI(int *V, int N);**

*** T buscaSequenciaR(int *V, int N);**

*** T buscaBinariaI(int *V, int N);**

*** T buscaBinariaR(int *V, int N);**

*** T buscaTernariaI(int *V, int N);**

*** T buscaTernariaR(int *V, int N);**

2.2 Ordenação

Dos algoritmos de ordenação, foram implementados os métodos *Bubble*, *Insertion*, *Selection*, *Quick*, *Merge* e *Decimal sort*, totalizando os 6 algoritmos implementados. A assinatura das funções pode ser vista à seguir:

*** void bubbleSort(int *V, int N);**

*** void insertionSort(int *V, int N);**

*** void selectionSort(int *V, int N);**

*** void quickSort(int *V, int N);**

*** void mergeSort(int *V, int N);**

*** void decimalSort(int *V, int N);**

2.3 Tipos Abstratos de Dados

Dos tipos abstratos de dados, foram implementados a lista ligada, o deque, a pilha e a fila, totalizando os 4 TADs implementados. Os TADs foram definidos em classes nas seguintes nomeclaturas:

*** Lista**

**** Deque***

*** Pilha**

*** Fila**

3. Testes realizados

Vários testes foram feitos para verificar a funcionalidade das implementações, dentre eles, podemos destacar:

3.1 Busca

Buscou-se em um vetor, criado aleatoriamente e ordenado de forma

crescente, por elementos presentes ou não no vetor, em caso positivo sua posição seria retornada e em caso negativo o valor padrão -1.

3.2 Ordenação

Geraram-se vetores aleatórios de tamanho pré-determinado e executamo-nos os algoritmos de ordenação a fim de verificar sua perfeita execução. Em todos os casos obtivemos resultados satisfatórios para as ordenações.

3.3 TADs

Foram instanciados objetos de TADs, seus atributos e métodos foram verificados e testados, para todos os casos houve resultado convincente.

4. Resultados

Os resultados obtidos mostraram-se aceitáveis ao que foi proposto, entretanto, não houve especialização de *templates* para outros tipos criados e/ou *strings/char*. Como trabalhos futuros, propõem-se a implementação destas especializações para garantir maior usabilidade da biblioteca gerada.

5. Dificuldades

Encontrou-se dificuldades para gerar a biblioteca genérica nos casos estáticos e dinâmicos, para *Linux* e *Windows*. Além do que, não foi possível realizar teste suficientes para atestar a funcionalidade das bibliotecas em sistemas *Windows*.