Nome: Vinicius Pereira Ximenes Frota NUSP: 11221967

Relatório EP1

Usando a Unit Test

O código da unit_test foi levemente modificada e algumas operações foram adicionadas. Ao executá-la, é possível fazer operações na tabela de símbolos:

‘minST’ devolve a primeira palavra em ordem alfabética.
‘delminST’ remove a primeira palavra em ordem alfabética.
‘getST <palavra>’ devolve o número de ocorrências da palavra no texto.
‘rankST <palavra>’ devolve o rank da palavra.
‘deleteST <palavra>’ apaga a palavra da tabela de símbolos.
‘selectST <rank>’ devolve a palavra com o rank <rank>.

Para eventuais debugs, as operações ‘printST’ e ‘insertST’ foram adicionadas.

Testes automáticos

Ao executar o script ‘conta’, testes automáticos são feitos para todas as estruturas de dados em dois textos diferentes. Ele testa os comandos getST, selectST, rankST e deleteST com os inputs presentes na pasta ‘entradas’. As saídas desses testes vão para a pasta ‘saidas’. Dessa forma, as saídas são reaproveitadas para a execução de um programa chamado ‘contador’ que conta os dados relacionados ao tempo de execução de cada função. A saída do ‘contador’ é direcionada para a pasta ‘testesDeTempo’, que contém um arquivo para cada função. Além disso, o script também usa o programa ‘testando’ para verificar se o resultado obtido pela estrutura de dado da iteração anterior é igual ao resultado obtido pela estrutura de dado da iteração atual.

Ao executar o ‘valgrindTeste’, ele apenas vai testar todas as estruturas de dados com os dois textos. A ideia desse script é apenas testar vazamento de memória em relação aos construtores e destrutores.

Ao executar o ‘valgrindRemocao’ um input (que está na pasta ‘entradas’) contendo operações ‘getST’ e ‘deleteST’ é executada. Assim como o script ‘conta’, há uma comparação da saída obtida da estrutura de dado da iteração atual pela estrutura de dado anterior. Além disso, o unit_test é executado com valgrind para teste de vazamento de memória.

Os arquivos de testes de tempo foram mantidos.