

Trabalho Prático 1

Vitor Hugo Lacerda Lana 2018072301

Professor: Carlos Henrique de Carvalho

7 de Outubro de 2019

1 Introdução

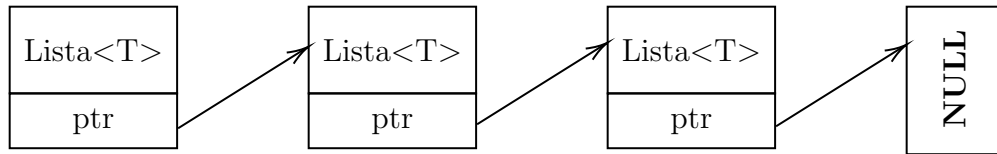
O trabalho apresentado abaixo foi desenvolvido após a proposta de otimização do tempo de medição de materiais do cientista Rick Sanchez em seu laboratório.

O objetivo principal do programa é informar para o usuário a quantidade mínima de operações que o mesmo deve realizar para medir uma certa quantidade Q de ml, a partir dos ml dos recipientes que o usuário informa para o software durante a execução.

O software do trabalho foi desenvolvido utilizando a linguagem C++.

2 Implementação

Para o desenvolvimento da solução foi utilizado uma TAD de lista dinâmica simples, nessa lista foi armazenado os valores inteiros que o usuário insere. Para o desenvolvimento da lista foi utilizado uma lista genérica, assim caso tenha necessidade de alterar o tipo de dado que deseja analisar é possível realizar alterações no programa sem muitas dificuldades.



A inserção na lista é realizada de forma a todo valor ser inserido no final da lista, em termos de processamento isso economiza tempo para a inserção, pois quando for procurar o valor para o usuário independente da lista está ordenada ou não é necessário procurar o valor da célula na lista. Além disso o algoritmo apenas insere na lista valores únicos, o usuário caso tenha dois recipientes com a mesma quantidade não terá diferenças no número de operações que o mesmo irá realizar, isso diminui o número de operações de comparação que o algoritmo realiza e diminui a memória utilizada pelo sistema.

A remoção de valores é feita informando o valor que deseja remover da lista, considerando que o usuário apenas irá remover valor de algum recipiente, a remoção é realizada na lista principal.

Para realizar as operação de busca do menor número de operações para encontrar o quantidade de ml que o usuário deseja informando os recipientes que o mesmo possui, é utilizado recursividade. Quando a função de resultado é chamada primeiro ela verifica se o quantidade de ml que o usuário deseja já está na lista de recipientes que o mesmo inseriu, caso esteja ela já retorna o valor para o usuário que o mesmo necessita realizar apenas uma operação. Caso a quantidade não esteja nessa lista, o programa chama a função novamente, repassando para ela a a lista original de recipientes do usuário, adiciona mais 1 a uma variável que armazena o número de operações e uma nova lista onde ele vai realizar as operações de soma e subtração de cada recipiente, caso o valor da subtração seja menor que 0, valor não é nem armazenado na nova lista, após todas as operações serem realizadas ele verifica se a quantidade que o usuário deseja está na nova lista, caso esteja o programa retorna o número de operações realizadas até o momento que é impresso para o usuário, caso não esteja ele chama a função novamente, repassando a lista original, e a nova lista para serem realizadas as operações novamente.

Os códigos para roda o sistema estão disponíveis no repositório https://github.com/vitorlana/ed/tree/master/rick_recip.

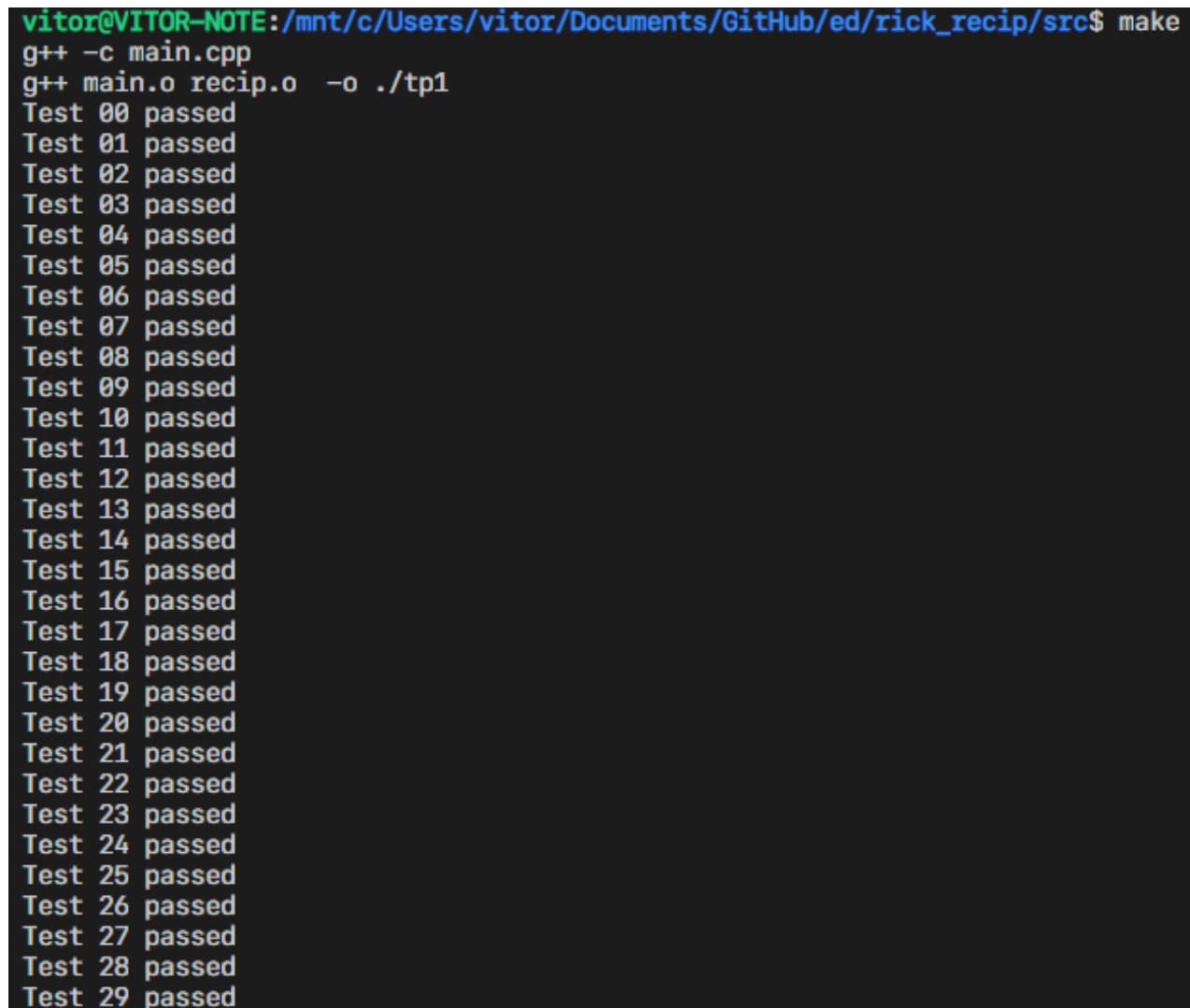
3 Instruções Compilação

Junto aos arquivos do programa segue um arquivo Makefile configurado para compilar os códigos do programa, e realizar os testes do mesmo.

Para realizar a compilação e os testes basta utilizando um sistema Linux (Utilizei para testes Ubuntu 18.04, rodando WSL), entrar no terminal dentro da pasta **src**, e digitar o comando **make**.

O programa irá compilar e executar todos os testes.

Segue abaixo print da tela com os testes e o comando.

A terminal window with a black background and white text. The prompt is 'vitor@VITOR-NOTE:/mnt/c/Users/vitor/Documents/GitHub/ed/rick_recip/src\$'. The command 'make' has been executed. The output shows the compilation of 'main.cpp' into 'main.o' and 'recip.o' into './tp1'. Following this, 30 tests are listed, each on a new line, all of which are marked as 'passed'.

```
vitor@VITOR-NOTE:/mnt/c/Users/vitor/Documents/GitHub/ed/rick_recip/src$ make
g++ -c main.cpp
g++ main.o recip.o -o ./tp1
Test 00 passed
Test 01 passed
Test 02 passed
Test 03 passed
Test 04 passed
Test 05 passed
Test 06 passed
Test 07 passed
Test 08 passed
Test 09 passed
Test 10 passed
Test 11 passed
Test 12 passed
Test 13 passed
Test 14 passed
Test 15 passed
Test 16 passed
Test 17 passed
Test 18 passed
Test 19 passed
Test 20 passed
Test 21 passed
Test 22 passed
Test 23 passed
Test 24 passed
Test 25 passed
Test 26 passed
Test 27 passed
Test 28 passed
Test 29 passed
```

Figura 1: Print do teste e Makefile

4 Conclusão

O trabalho foi de grande importância para desenvolvimento e utilização dos conceitos estudados de TADs funções recursivas e outros assuntos vistos durante a matéria de estrutura de dados.

Uma grande dificuldade foi desenvolver e compreender a utilização de uma lista genérica, esse conceito de uma lista mais genérica senti que foi de grande importância para aprendizados e modularização e reutilização de códigos.

5 Bibliografia

<https://www.learncpp.com/cpp-tutorial/67-introduction-to-pointers/>
<https://www.learncpp.com/cpp-tutorial/131-function-templates/>
<http://www.cplusplus.com/doc/tutorial/dynamic/>