



UF *m* G

Universidade Federal de Minas Gerais
Departamento de Ciência da Computação
Estrutura de Dados

Trabalho Prático 2

Vitor Hugo Lacerda Lana 2018072301

Professor: Carlos Henrique de Carvalho

12 de Novembro de 2019

1 Introdução

O trabalho apresentado abaixo foi desenvolvido após a proposta de otimização do tempo gasto por Rick Sanchez para realizar as visitas nos planetas.

O principal objetivo do TP é informar para o Rick quais são os planetas que o mesmo deve visitar cada mês e qual a ordem de visitação dos planetas em ordem alfabética.

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

2 Implementação

Para o desenvolvimento da solução foi utilizado um vetor de ponteiros alocado dinamicamente ao tamanho da entrada do usuário. Para armazenar todas as informações de uma forma simples, foi utilizado uma struct. Para solucionar o problema apresentado, foram desenvolvidas funções que trabalham com o vetor de ponteiros. Para começar vamos falar da função de inserção, a mesma recebe a quantidade de planetas, e o número máximo de caracteres do nome de cada planeta e realiza a inserção, as próximas duas funções são utilizadas para ordenação.

A primeira função para ordenar é utilizada para ordenar quanto tempo cada planeta necessita para realizar a visita, pela necessidade da função ser $O(n \log(n))$ e ser estável foi utilizado o método de ordenação MergeSort. Após realizar a ordenação por tempo necessário de cada planeta, o algoritmo chama a função organiza agenda, esse parte do algoritmo, recebe o vetor de planetas que foi ordenado pela função anterior, o mesmo verifica o tempo disponível máximo inserido pelo usuário e organiza em meses a quantidade máxima de planetas que o mesmo consegue visitar, além de inserir na struct do planeta o mês que o mesmo deverá ser visitado.

Após o sistema organizar os meses para visitação, o mesmo ainda não está ordenado em ordem alfabética, pra isso chamamos um algoritmo RadixSort, o mesmo cumpre a necessidade de ter complexidade $O(n * x)$ para ordenar em ordem alfabética.

Para finalizar as funções do algoritmo, para ele imprimir para o usuário a ordem de planetas e quantidade de meses necessários para visitar todos é chamada a função imprime ordenado, para economizar espaço de memória a mesma gasta mais processamento para imprimir todos em ordem de mês que deve ser visitado, tempo que será gasto no planeta e o nome do mesmo.

Como biblioteca importante foi utilizada a biblioteca string.h, ela foi necessária para manipulação dos nomes dos planetas.

Os códigos para executar o sistema estão disponíveis no repositório https://github.com/vitorlana/ed/tree/master/tp2/rick_trip.


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 na WSL), entrar no terminal dentro da pasta **src**, e digital o comando **make**.

O programa irá compilar e executar todos os testes.

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



```
vitor@VITOR-NOTE:/mnt/c/Users/vitor/Documents/GitHub/ed/tp2/rick_trip/src$ make
g++ -c -o base.o base.cpp
g++ -Wall -Wextra main.cpp base.o -o ./tp2
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
```

Figura 1: Print do teste e Makefile

4 Conclusão

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

Uma grande dificuldade foi desenvolver e compreender a utilização de vetor de ponteiros para implementação dos métodos de ordenação.

5 Bibliografia

<https://www.geeksforgeeks.org/merge-sort-for-linked-list/>

<http://www.cplusplus.com/forum/general/39884/>

<https://www.geeksforgeeks.org/core-dump-segmentation-fault-c-cpp/>

https://opendatastructures.org/ods-cpp/11_2_Counting_Sort_Radix_So.html