Trabalho Prático 1: O problema de medição de Rick Sanchez

Gabriel Chaves Ferreira – 2018046700

1. Introdução

O problema é criar um algoritmo capaz de resolver o problema de medição. Este trata-se de ter recipientes de x ml, e querer medir y ml utilizando o menor número de combinações possíveis com os recipientes x disponíveis. Para solucionar o problema, foi utilizado um algoritmo de bottom up, resolvendo primeiro os problemas mais simples e fáceis, e escalando para os mais difíceis posteriormente.

Além disso, o programa foi escrito em C++, primeiramente não implementei as listas com código puro, utilizando uma biblioteca já pronta para que tivesse uma visão geral da implementação necessária para resolver o problema. Posteriormente as listas foram implementadas com código, utilizando ponteiros e strucs, para mais detalhes visite o meu github: https://github.com/gabrielchaves7/trabalhoED.

2. Implementação

Implementei as listas em C++, utilizando uma lista duplamente encadeada, com um inicio e um fim. Existe uma função geral para iniciar as listas, e outra função para cada operação básica de uma lista: inserção no fim, inserção em qualquer posição, remoção e limpeza da lista. Também tive que criar uma função para realizar um deep copy das listas, visto que quando eu utilizava o operador "=" o compilador realizava um shallow copy, e não era o resultado que eu esperava.

Além disso, também foi implementada uma função para verificar se a lista já possui um elemento, que é a base que eu utilizo para achar a quantidade desejada e para evitar que combinações repetidas sejam inseridas na lista.

Por fim, não vi necessidade em implementar classes e orientação à objetos, pois não vi vantagem em ter disponível polimofirsmo e o encapsulamento não se fez necessário pois as variáveis deveriam ser expostas para todo o código.

3. Instruções de compilação e execução

Abra o arquivo no seu compilador de preferência e rode o comando make. Após isso é só executar o executável que sairá com o nome de "output".

4. Análise de complexidade

A função principal (que calcula quantas operações são necessárias para medir), tem como complexidade $O(n^3)$. Pois são utilizados 3 "fors", um dentro do outro.

5. Conclusão

O principal problema enfrentado foi descobrir porque quando eu igualava duas listas não era realizado um deep copy. Até eu descobrir a diferença entre um deep copy e um shallow copy e que o C++ por padrão realiza um shallow copy quando você iguala

duas structs um tempo considerável foi perdido.

6. Bibliografia

Acesse o meu github: https://github.com/gabrielchaves7.