Relatório – A sorveteria dos horrores

Arthur Viegas e Henrique Ramires

Escola Politécnica – PUCRS

*Resumo: Este artigo descreve uma alternativa para solucionar o problema proposto na disciplina de Algoritmos e Estrutura de Dados II, que consiste em descobrir o número de combinações possíveis de copinhos de sorvete segundo as regras de uma sorveteria. A modelagem do problema e o processo da solução é apresentado, juntamente com o pseudocódigo dos algoritmos mais importantes. Por fim, serão apresentados os resultados obtidos nos seis casos de teste disponibilizados pelo professor.*

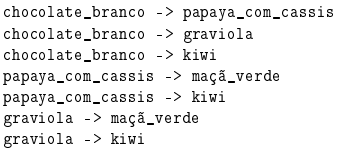
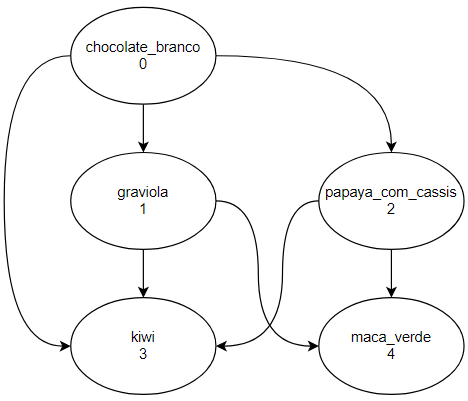
**Enunciado do problema**

No contexto do problema, meu primo abriu uma sorveteria, mas impôs algumas regras sobre os sorvetes que ele vende, sendo estas:

1. Na sorveteria só são vendidos copinhos com 2 ou 3 bolas de sorvete.
2. É proibido misturar um sabor forte com um sabor fraco (Exemplo: Chocolate com Iogurte. Para saber quais sabores são fortes ou fracos há uma lista).
3. Não é permitido repetir sabores, para que as pessoas provem mais sabores.

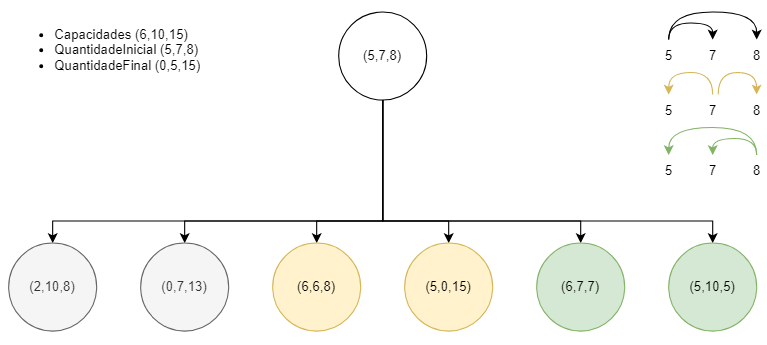
A família está preocupada com as regras que ele impôs, e acham que isso pode limitar o número de pedidos da sorveteria, então me foi pedido para criar um programa que cheque o número de combinações possíveis de copinhos de sorvete.

Abaixo um exemplo da lista de proibições da sorveteria junto de um diagrama da lista para facilitar a visualização.

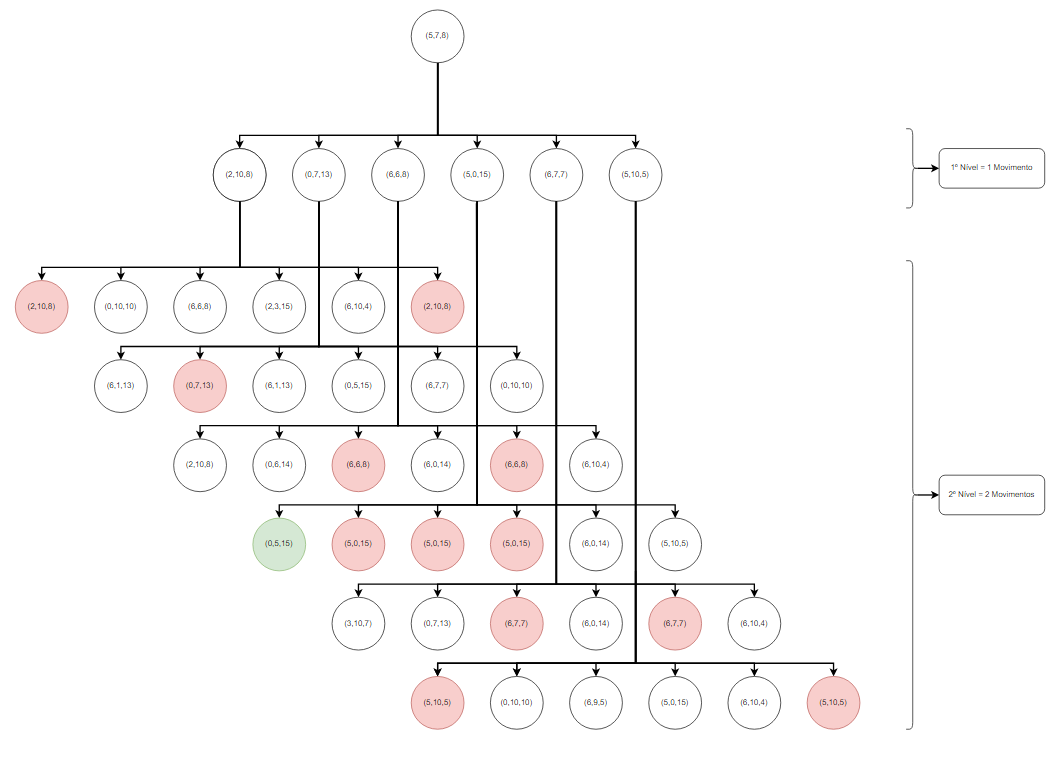
**Etapas da solução, apresentando exemplos e algoritmos**

Desde o primeiro momento do desenvolvimento do processo, tentei resolver através de uma Árvore Não-Binária. A ideia principal é que a partir de um nodo original, sejam criados 6 novos nodos a partir da movimentação de toda a água possível de um JarroOrigem para um JarroDestino, conforme a ilustração abaixo:



Nesta próxima etapa cada nodo é adicionado em uma lista de nodos e um método checa se algum dos nodos atuais é o Nodo que contém os jarros com a quantidade final desejada.

Seguindo este exemplo, o nodo final ainda não foi encontrado, então, para cada nodo criado no nível anterior da árvore nós criamos mais 6 nodos, através de recursão.



Aqui neste exemplo, há diversos nodos que não serão adicionados na lista de novos nodos, pois eles são nodos inválidos, sendo alguns nodos ou repetidos ou não sendo possíveis de realizar o despejo da água.

Como atingimos o nodo desejado em 2 movimentos, então após o programa comparar o nodo criado com o final, o programa se encerra.

Aqui abaixo alguns algoritmos do programa:

**Main do programa:**

****

Atualmente a função main do programa, lê um arquivo de texto com 3 linhas (através do BufferedReader), quebra as linhas em vetores de Strings e instancia os jarros com as informações dos vetores de Strings. No final do programa, ele escreve o Nodo Final desejado em um arquivo de texto

**Algoritmo de despejo de água:**

O despejo de água funciona com a união do método do jarro de origem enviando a água contida nele atualmente, para um método receptor, no jarro de destino, que faz os cálculos das capacidades e devolve a quantidade excedente (se houver), para o jarro de origem.



**Resultados dos casos de teste**

No momento atual não consegui finalizar o algoritmo que faz tudo automaticamente, sendo possível atingir o nodo final apenas transferindo a água manualmente, contudo já possuo o algoritmo para a execução automática

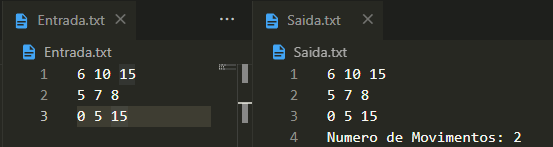
Sendo ele:

1. Cria Nodo Pai
2. Gera 6 Nodos.

* Nodo1 = Jarro1.despejaAgua(Jarro2)
* Nodo2 = Jarro1.despejaAgua(Jarro3)
* Nodo3 = Jarro2.despejaAgua(Jarro1)
* Nodo4 = Jarro2.despejaAgua(Jarro3)
* Nodo5 = Jarro3.despejaAgua(Jarro1)
* Nodo6 = Jarro3.despejaAgua(Jarro2)

1. Adiciona Nodos em Lista de Novos Nodos
2. Checa Se um dos Nodos Novos é a solução (Se não: Contador de Movimentos++)
3. Volta para a segunda etapa tendo o Nodo1 como “Nodo Pai”

Entrada e Saída com o caso 1 manualmente:



**Conclusões**

Gostei de desenvolver o programa, a ideia dele é muito boa, contudo, foi infeliz de minha parte não conseguir terminar o construtor da árvore, mas fora isto o trabalho foi bom para o meu desenvolvimento profissional e pessoal. Mesmo após a entrega do trabalho, continuarei o desenvolvimento do programa, para fins de estudo, e deixo aqui o [link para o repositório](https://github.com/ArthurViegas01/Trabalho1Alest2).