

Relatório de Desenvolvimento – Trabalho 2

Algoritmos e Estruturas de Dados II

André Bins, Érico dos Santos, Lucas Bandeira, Marcos Ferreira, Vermon Aguiar

20 de novembro de 2022

1 Descrição do Problema

O desafio proposto para o trabalho 2 da disciplina de Algoritmos e Estruturas de Dados II foi o desenvolvimento de um algoritmo para fundamentar a composição de pedidos em uma sorveteria, a qual possui uma série de fatores limitantes em relação às possibilidades de sabores na criação de diferentes copinhos de sorvete. Assim, são definidas algumas regras para a criação de um copinho pelo cliente, conforme segue:

- A sorveteria tem copinhos para 2 e 3 bolas de sorvete;
- Não é permitido ao cliente colocar um sabor forte sobre um sabor suave;
- Pedidos com sabores repetidos não são aceitos pela sorveteria.

Dessa forma, considerando as regras definidas pela sorveteria e uma lista de sabores mais fortes e mais suaves, o algoritmo deve ser capaz de considerar essas informações e produzir um resultado equivalente à quantidade de copinhos de 2 e 3 sabores que podem ser servidos.

2 Descrição da Solução

O algoritmo proposto foi desenvolvido na linguagem de programação Java, utilizando como estrutura de dados para modelar o problema um Grafo Direcionado, representando as relações de quais sabores são mais fortes do que outros, que são utilizadas como restrições para as combinações de sabores. Para modelar o Grafo Direcionado do problema, optamos por utilizar uma matriz de adjacências, para facilitar e tornar mais rápido a verificação de se existe uma aresta conectando dois vértices, o que vai ser muito utilizado em nosso algoritmo. Os vértices representam os sabores de sorvete, e as arestas as relações de quem-é-mais-forte-do-que-quem. Para mantermos a informação de qual índice (valor inteiro) da matriz representa qual sabor, utilizamos um HashMap com a String do nome do sabor como chave e um número inteiro como valor.

A estrutura do código é composta simplesmente do método main, onde todo algoritmo é desenvolvido. Inicialmente, o arquivo texto contendo o caso de teste desejado é lido, e a representação do Grafo é criada, utilizando a matriz de adjacências. Após isso, foi criado o Algoritmo 1, que será aprofundado na Seção dos Pseudocódigos, mas que inicialmente percorre a matriz, construindo todas relações entre sabores que ainda não existiam no Grafo. Por exemplo, se inicialmente existir uma aresta de flocos para chocolate, e uma de chocolate para menta, deve ser criada uma aresta de flocos para menta, pois flocos é mais forte do que menta, e é uma combinação possível. Para garantir que todas as novas relações necessárias sejam criadas, esse processo é feito duas vezes. Depois disso, a contagem de combinações possíveis é feita em duas partes: na primeira são contadas as combinações entre dois sabores, relações estabelecidas por uma única aresta, e depois são contabilizadas as relações entre três sabores, que são estabelecidas por duas arestas.

2.1 Pseudocódigo

Abaixo, será apresentado o pseudocódigo do algoritmo desenvolvido para calcular a quantidade de copinhos de sorvete de 2 ou 3 sabores, a partir da leitura do arquivo de casos de teste.

Algorithm 1 Algoritmo para cálculo da quantidade de combinações

```
1: while não percorreu 2 vezes a matriz do
2:   for each vértice  $x$  do
3:     for each vértice  $j$  do
4:       if existe uma aresta de  $(x,j)$  then
5:         for each vértice  $a$  do
6:           if existe uma aresta  $(j,a)$  then
7:              $a$  é adicionado na pilha
8:           while pilha não está vazia do
9:             Desempilha um valor  $z$ 
10:            Cria uma aresta  $(x, z)$ 
11: for each vértice  $x$  do
12:   for each vértice  $j$  do
13:     if existe uma aresta de  $(x,j)$  then
14:       Incrementa contador de copinhos com 2 sabores
15: for each vértice  $x$  do
16:   for each vértice  $j$  do
17:     if existe uma aresta de  $(x,j)$  then
18:       for each vértice  $a$  do
19:         if existe uma aresta de  $(j,a)$  then
20:           Incrementa contador de copinhos com 3 sabores
```

3 Resultados dos Casos de Teste

A tabela 1 abaixo apresenta os resultados obtidos ao utilizar 6 casos de teste, contendo uma lista de hierarquia de sabores indicando a relação de sabores mais fortes e mais suaves. Assim, está elencado, para cada caso de teste, o número de sabores existentes, o número de combinações contendo 2 e 3 sabores, e, por fim, o número de operações realizadas para concluir o algoritmo.

Tabela 1: Resultados dos casos de teste

Caso de teste	Nº sabores	Combinações de 2 sabores	Combinações de 3 sabores	Nº de operações
1	9	14	11	723
2	19	83	155	5811
3	30	257	929	26952
4	40	417	1633	54223
5	50	865	7058	137571
6	60	1245	12050	236670

Analisando os resultados dos casos de teste, observamos que, sendo V o número de sabores existente no caso de teste, o número de operações realizadas pelo algoritmo em cada caso é de aproximadamente V^3 operações.

3.1 Visualização da Execução - Caso de Teste 1

O algoritmo utilizado roda para todos os vertices do grafo. Na figura 1, é mostrado o procedimento de apenas um passo com o vértice focos em execução. O algoritmo começa no vértice focos e verifica se possui outros vértices adjacente, nesse exemplo, é possível constatar que o vértice tutti frutti é adjacente a focos. Então, o algoritmo verifica os vértices que são adjacentes ao vértice tutti frutti e os adiciona como adjacentes ao vértice focos, repetindo esse processo para todos os vértices do grafo.

A figura 2 mostra todas as novas arestas adicionadas ao grafo após rodar o algoritmo. Realizando então a contagem das arestas no grafo, é possível descobrir o número de combinações possíveis de copinhos com 2 sabores, pois as arestas representam quais sabores são mais fortes do que outros. E para descobrir as combinações possíveis com 3 sabores, a única mudança necessária é que ao verificar que existe uma aresta de um sabor x para y , o algoritmo contabiliza os adjacentes de y .

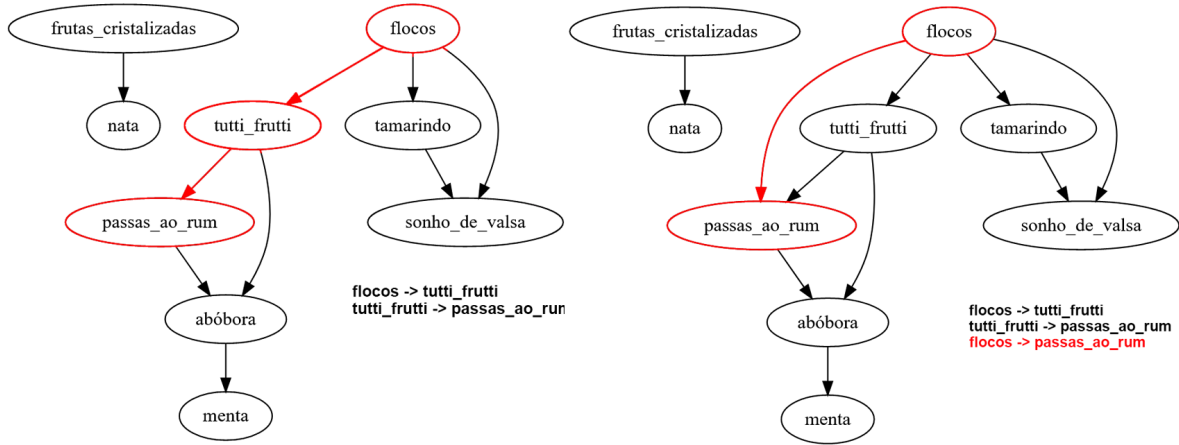


Figura 1: Visualização do grafo do caso de teste

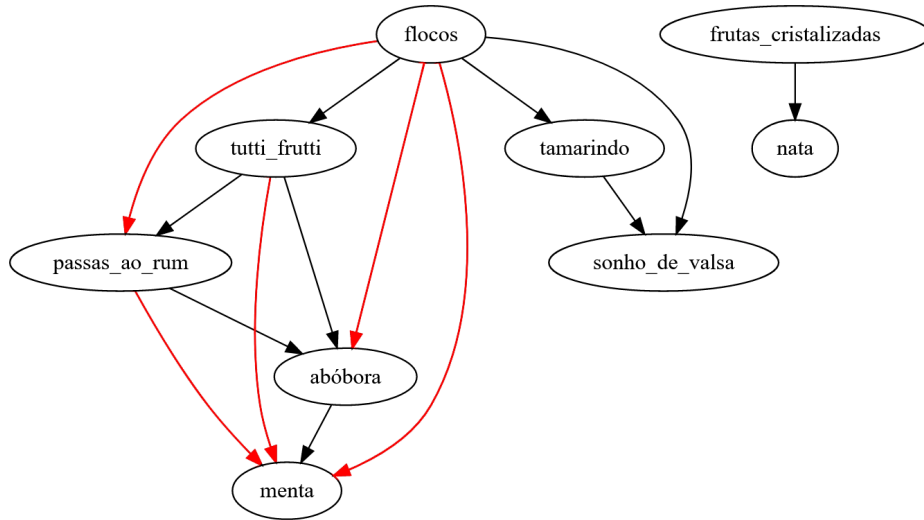


Figura 2: Visualização do grafo do caso de teste 1 com as novas arestas adicionadas

4 Conclusões

O algoritmo desenvolvido atingiu os resultados esperados, dado que em todos os casos de teste passados pelo professor, o programa foi capaz de considerar a hierarquia de sabores mais fortes e mais suaves, encontrando ao final a quantidade total de combinações possíveis indicada para cada um dos casos. Além disso, destaca-se que o programa poderia ser mais eficiente, pois considerando os seis casos de teste e o número V de sabores em cada caso, foram realizados, para cada caso aproximadamente, V^3 iterações.