

Trabalho Prático 02 – AEDS 1 – Em Trio

Professora: Thais R. M. Braga Silva

Valor: 10 pontos

Data de Entrega: 04/11/2022

Forma de Entrega: PVANet Moodle (formato .zip ou .tar.gz)

INTRODUÇÃO

O objetivo deste trabalho prático é permitir a avaliação do impacto causado pelo desempenho dos algoritmos em sua execução real. Vimos em sala de aula que existem problemas e algoritmos de complexidade exponencial, chamados de intratáveis. Nesses casos, os programas, ao serem executados, podem demorar uma quantidade de tempo não razoável para encontrar uma solução, dependendo do tamanho da entrada. Vamos observar, portanto, como isso ocorre na prática.

Para tanto, cada trio fará uma implementação para o **Problema das Bolinhas de Natal (PBN)**. Esse é um problema intratável, pois sua solução exata somente é possível através do cálculo e avaliação de todas as possíveis saídas, o que chamamos de força bruta. Neste trabalho, essa implementação deverá ser executada para diferentes tamanhos de entrada, e o tempo gasto para que o programa termine, em cada caso, deverá ser medido por meio de comandos do sistema operacional.

PROBLEMA DAS BOLINHAS DE NATAL (PBN)

O Natal está chegando e todos querem ter uma árvore de Natal bem colorida e decorada em casa. Para isso, é necessário espalhar as bolinhas de Natal pela árvore, de modo que **bolinhas adjacentes** (que estejam uma do lado da outra ou uma acima/abaixo da outra) **não possuam a mesma cor**. Isso garante que todas as cores de bolinhas disponíveis estarão bem espalhadas por toda a árvore. Podemos formular e resolver esse problema computacionalmente. Este é o chamado Problema das Bolinhas de Natal (PBN).

O PBN é um problema clássico da literatura que pode ser definido como se segue. **Dado um conjunto de K cores de bolinhas** de Natal e uma árvore de Natal **contendo N espaços/locais para pendurá-las**, encontrar uma distribuição das bolinhas nos espaços **tal que espaços adjacentes** (que estejam um ao lado do outro ou um acima/abaixo do outro) **não contenham bolinhas da mesma cor**.

Em linhas gerais, **esse problema recebe como entrada K cores, o número N de espaços para se pendurar bolinhas na árvore de Natal e o conjunto de adjacências entre pares de espaços E_i e E_j** (considere que a adjacência entre E_i e E_j representa a proximidade entre eles na árvore, sendo que a adjacência entre E_i e E_j é a mesma daquela entre E_j e E_i). Os espaços são identificados por números, que vão de 0 a N-1. Assim, o conjunto de espaços é dado por $\{E_0, E_1, E_2, \dots, E_{N-1}\}$. O conjunto de adjacências informa, para cada espaço E_i quais são os espaços adjacentes a ele.

O Problema das Bolinhas de Natal consiste em determinar se, dadas as K cores, existe um conjunto de distribuição de cores pelos espaços, tal que espaços adjacentes não utilizem a mesma cor.

Uma solução do PBN é representada por meio de um arranjo com repetição de tamanho N das K cores. Cada i -ésima posição no arranjo representa o i -ésimo espaço na árvore de Natal. A cor atribuída à esta posição seria a cor da bolinha de Natal a ser utilizada naquele espaço. Por exemplo, dada uma árvore de natal com 5 espaços para pendurar bolinhas e 3 cores de bolinhas diferentes (Azul - A, Verde - V, Roxo - R), uma solução do PBN nesse caso poderia ser [A, V, V, R, A] considerando que o espaço E_0 não seja adjacente ao E_4 e o espaço E_1 não seja adjacente ao E_2 .

SOLUÇÃO EXATA PBN

Existem diversas possíveis implementações para o PBN. Entretanto, as mais utilizadas requerem recursos de programação que ainda não foram estudados por vocês. Dessa forma, adotaremos uma estratégia mais direta, visto que o objetivo principal do trabalho é a avaliação de desempenho, e não o desenvolvimento do algoritmo. Cada dupla deverá implementar um programa, em linguagem C, para o PBN da seguinte forma:

- Procurar na Web um algoritmo de arranjo com repetição para um conjunto de K elementos, N a N . Esse algoritmo deve, portanto, produzir todos os possíveis arranjos de tamanho N entre os K elementos. Você deverá utilizar a linguagem C. Caso o algoritmo encontrado esteja em outra linguagem de programação, vocês deverão convertê-lo para a linguagem C.
- Para armazenar as adjacências entre os N espaços da árvore de Natal, utilizar uma matriz M [$N \times N$]. Dados dois espaços i e j na árvore, caso eles sejam adjacentes (ocorram lado a lado ou um acima/abaixo do outro), adicionar o valor 1 nas posições $M[i][j]$ e $M[j][i]$. Colocar 0 caso contrário. As posições da diagonal principal $M[i][i]$ da matriz devem receber -1 e devem ser desconsideradas.
- Utilize o algoritmo de arranjo para gerar todos os arranjos com repetição de tamanho N para os elementos do conjunto K de cores de bolinhas.
- Para cada arranjo resultante do passo anterior, utilize a matriz M para verificar se existem espaços adjacentes usando bolinhas de cores iguais. Para tanto, comece do primeiro elemento do arranjo. Verifique, pela matriz M , quais são os espaços adjacentes a ele. Em seguida, verifique se algum desses espaços está utilizando a mesma cor. Em caso positivo, esse arranjo não é uma solução válida e pode ser descartado, avançando-se para o próximo arranjo. Em caso negativo, avance para o próximo elemento do arranjo fazendo a mesma verificação. Caso chegue ao final do arranjo e verifique que nenhum espaço possua a mesma cor de seus espaços adjacentes, uma solução válida foi encontrada e o PBN foi resolvido.
- Execute o programa para entradas com valor $K = 2$ (ou seja, existem duas cores de bolinhas de Natal) e N começando em 10. Em seguida, vá dobrando o valor de N . Ao executar o programa, utilize uma ferramenta para medição do tempo de execução, como o comando *time* do Unix.

- Faça um relatório final utilizando o mesmo formato disponibilizado para o TP1. Não se esqueça de explicar como funciona o algoritmo para arranjo com repetição utilizado, de onde foi obtido, como o seu programa usa esse algoritmo e como ele faz as verificações sobre cada arranjo gerado. **Descreva também no relatório qual a configuração da máquina usada para rodar o trabalho.** Por fim, o relatório também deverá conter os resultados dos tempos de execução para os valores de N conforme demandado acima. Lembrem-se de colocar no texto os nomes e matrículas dos 3 integrantes do trio.
- Coloque no seu relatório uma análise sobre os tempos obtidos para cada valor de N testado e, ao fim, faça um gráfico que ilustre o ocorrido. Comente sobre o último valor de N testado por vocês e justifique o motivo pelo qual vocês não avaliaram valores mais altos.
- Um arquivo contendo o código fonte e o relatório final deverá ser entregue (por um único membro do grupo) até a data limite pelo PVANet Moodle.
- Para testar a implementação do problema produzido, você deverá receber um arquivo como entrada, o qual irá conter os conjuntos de adjacências para uma dada execução do PBN. No exemplo abaixo, as linhas descrevem as adjacências da seguinte forma: o primeiro valor informa um elemento do arranjo, sendo que os seguintes são os elementos adjacentes àquele primeiro. Exemplo:

0 1 3

1 0 2

2 1 4

3 0 4

4 2 3

Matriz de adjacência preenchida após leitura da entrada:

| I\J | 0 | 1 | 2 | 3 | 4 |
|-----|----|----|----|----|----|
| 0 | -1 | 1 | 0 | 1 | 0 |
| 1 | 1 | -1 | 1 | 0 | 0 |
| 2 | 0 | 1 | -1 | 0 | 1 |
| 3 | 1 | 0 | 0 | -1 | 1 |
| 4 | 0 | 0 | 1 | 1 | -1 |