



SERVIÇO PÚBLICO FEDERAL · MINISTÉRIO DA EDUCAÇÃO

UNIVERSIDADE FEDERAL DE VIÇOSA · UFV

CAMPUS FLORESTAL

Trabalho 2 - AEDS 1

Problema das Bolinhas de Natal

Matheus Nascimento Peixoto [EF04662]

Matheus Nogueira Moreira [EF04668]

Pedro Augusto Martins [EF04692]

Florestal - MG

2022

Sumário

1. Introdução	3
2. Organização	3
3. Desenvolvimento	4
3.1 - Arquivos de entrada	4
3.2 - Arquivos ADJACENCIAS.H e ADJACENCIAS.c	5
3.3 - Arquivos ARRANJO.h e ARRANJO.c	6
3.4 - Arquivos Funcoes.h e Funcoes.c	6
4. Execução	6
5. Resultados	6
7. Referências	8

1. Introdução

Objetivando o desenvolvimento prático da matéria estudada em sala de aula, o “Problema das Bolinhas de Natal” visa demonstrar o impacto causado por algoritmos conhecidos como intratáveis, isto é, que não possuem um algoritmo cuja sua função de complexidade seja polinomial para resolvê-lo. No caso, este é um algoritmo de complexidade exponencial.

Recebendo um arquivo de texto, o programa precisa transformar as informações recebidas nesse arquivo em uma matriz chamada de “Matriz de Adjacências”. Esta matriz recebe o valor inteiro “-1” para a sua diagonal principal, o valor “1” para as posições de adjacência e 0 para as restantes.

A partir disso, serão gerados arranjos do tipo “Arranjo com Repetição” a fim de descobrir possíveis soluções para o problema.

2. Organização

O trabalho foi organizado em 3 arquivos cabeçalhos com suas respectivas funções, uma main, um arquivo makefile, para auxiliar na execução e possui 5 opções de entradas já testadas e uma denominada “entrada_de_teste_do_usuario.txt” que está vazia, a fim de que o usuário que for testar o programa insira o conteúdo que deseja testar neste arquivo. Abaixo, na imagem 1, será possível observar essa organização.

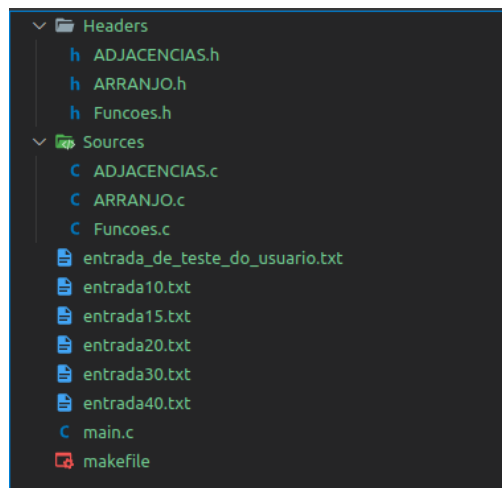


Imagem 1: Organização dos arquivos do trabalho

3. Desenvolvimento

Para explicar como foi realizado o desenvolvimento deste Trabalho Prático 02 serão divididos em tópicos os arquivos e explicando sua importância e contribuição para a execução do programa.

3.1 - Arquivos de entrada

Foram desenvolvidos alguns arquivos de entrada. A seguir estão as matrizes de alguns destes arquivos.

$N = 10$

-1	0	0	0	0	0	0	0	1	1
0	-1	1	1	0	0	0	0	0	0
0	1	-1	1	0	0	0	1	0	0
0	1	1	-1	0	0	0	0	0	0
0	0	0	0	-1	1	0	0	0	1
0	0	0	0	1	-1	0	0	0	0
0	0	0	0	0	0	-1	0	1	1
0	0	1	0	0	0	0	-1	0	0
1	0	0	0	0	0	1	0	-1	0
1	0	0	0	1	0	1	0	0	-1

0 9 8
1 3 2
2 1 7 3
3 2 1
4 5 9
5 4
6 8 9
7 2
8 0 6
9 0 4 6

Imagem 2: entrada10.txt

$N = 15$

-1	0	0	0	0	0	1	1	1	0	0	0	0	0	0
0	-1	1	0	0	1	0	0	1	0	0	0	0	0	0
0	1	-1	0	1	0	0	0	0	0	0	1	0	0	0
0	0	0	-1	0	1	1	0	0	0	0	0	0	0	0
0	0	1	0	-1	0	0	0	1	0	0	0	0	0	0
0	1	0	1	0	-1	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	-1	1	0	0	0	0	0	0	0
1	0	0	0	0	0	1	-1	0	0	0	0	0	0	1
1	1	0	0	1	0	0	0	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	-1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	1	-1	1	0	0	0
0	0	0	0	0	0	0	0	0	1	1	-1	0	0	0
0	0	1	0	0	0	0	0	0	0	0	-1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	1	-1	0	0
0	0	0	0	0	0	0	1	0	0	0	1	0	-1	0

0 6 7 8
1 2 5 8
2 1 4 12
3 5 6
4 2 8
5 1 3
6 0 3 7
7 0 6 14
8 0 1 4
9 10 11
10 9 11
11 9 10
12 2 13 14
13 12
14 7 12

Imagem 3: entrada15.txt

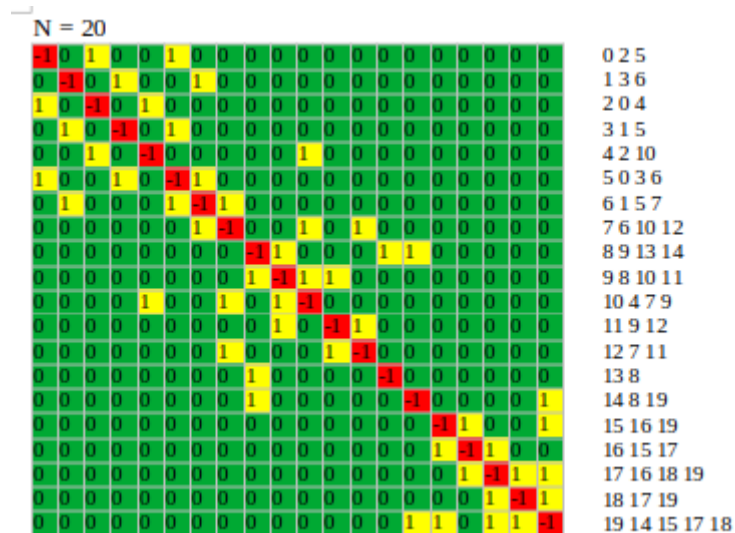


Imagem 4: entrada20.txt

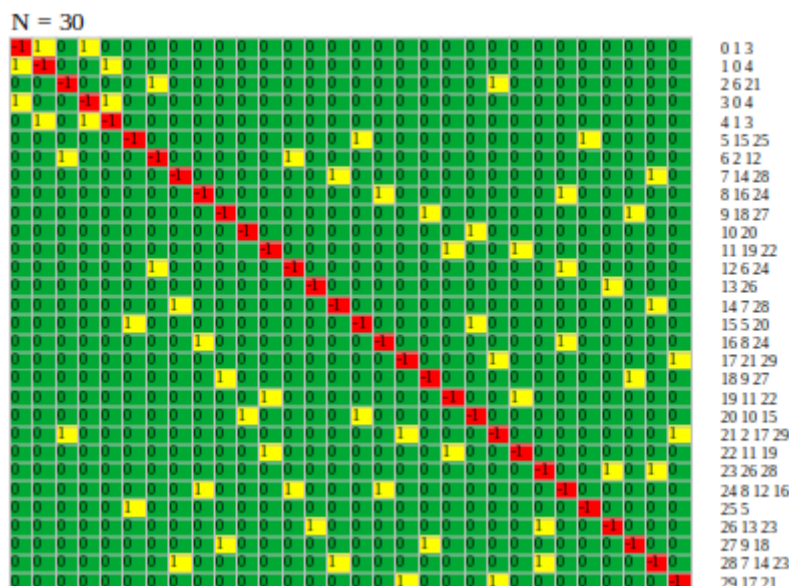


Imagem 5: entrada30.txt

3.2 - Arquivos ADJACENCIAS.H e ADJACENCIAS.c

Este arquivo tem como função receber o arquivo de texto e, através da leitura de suas linhas, saber identificar a quantidade de linhas que este possui, criando uma matriz que recebe -1 em sua diagonal principal, o valor 1 para as posições identificadas pela leitura do .txt e o valor 0 para as demais posições.

Após a criação da Matriz de Adjacências ela é impressa na tela.

Ao final do algoritmo, ele chama a função "ARRANJO", passando a ela como parâmetros o "n", isto é, o número de linhas que encontrou enquanto lia o arquivo de entrada e consequentemente o número de linhas e colunas da matriz, bem como a própria matriz.

3.3 - Arquivos ARRANJO.h e ARRANJO.c

Estes arquivos, como dito na subseção anterior, recebem como parâmetro um tamanho “n” e a “Matriz de Adjacências”.

Este algoritmo foi fortemente baseado no mesmo que pode ser encontrado no item 2 da seção Referências. Sua função é formar arranjos de tamanho “n” entre os valores “A” e “B”.

Para sua utilização no Trabalho, a cada arranjo gerado este é comparado com a “Matriz de Adjacências”. Com o objetivo de descobrir arranjos que possivelmente são condizentes com cada linha da matriz foi criado o contador “contador”. A cada arranjo gerado é comparado, caso a posição da matriz possua valor 1 é conferido se a posição [x] do arranjo e sua posição [j] apresentam valores iguais. Caso positivo ($str[x] = str[j]$) o programa recebe um “break”. Caso negativo, o “contador” é incrementado em 1. A mesma coisa acontece quando a posição da matriz a ser comparada é -1 ou 0.

Quando o contador for igual a “n” significa que aquele arranjo passou por toda a linha da matriz e apresentou um resultado positivo para aquela determinada linha, imprimindo-o na tela.

3.4 - Arquivos Funcoes.h e Funcoes.c

Estes arquivos apenas chamam as demais funções do programa, a fim de que o arquivo main fique mais organizado e limpo, apresentando um menu e executando a operação que lhe for exigida.

4. Execução

Para a execução deste projeto existe o arquivo “makefile” para facilitar.

Basta digitar no terminal:

```
make all [Pressione Enter]
```

Posteriormente, caso esteja utilizando um computador Linux:

```
make Linux [Pressione Enter]
```

Ou, caso esteja em uma máquina com Windows, insira:

```
make Windows [Pressione Enter]
```

Após isso, com o programa já em execução, basta digitar o tamanho do N desejado, sendo uma das seguintes opções: 10, 15, 20, 30, 40, 99 para a execução do arquivo de entrada “entrada_de_teste_do_usuario.txt” e 0 para parar a execução.

5. Resultados

Como resultado aparecem alguns arranjos que foram tidos pelo programa como possivelmente compatíveis com determinada linha da matriz.

A saída “O arranjo [“-----”] possivelmente serve para a linha [“-”] da matriz!” indica que este arranjo passou pela linha “-” da matriz e o contador recebeu um valor maior do que o “n” repassado, indicando que este é um possível resultado para esta determinada linha.

Alguns arquivos de entrada foram testados. A seguir serão apresentados os arquivos e suas respectivas médias de tempo.

Arquivo “entrada10.txt”: aproximadamente 1 segundo;

Arquivo “entrada15.txt”: aproximadamente 1 segundo;

Arquivo “entrada20.txt”: em média 07 segundos (*);

Arquivo “entrada30.txt”: em média 430 segundos (*);

Arquivo “entrada40.txt”: sua execução extrapolou 12 horas (**).

Devido ao elevado aumento do tempo de execução deste programa, valores maiores não foram testados.

(*)-Observação 1: Alguns testes apresentaram valores de tempo inconstantes, apesar de se tratar do mesmo arquivo. Um exemplo disso foi um teste da entrada com $N = 20$, o qual chegou a durar 16 minutos, quando o tempo normal obtido nos demais testes figurou entre 5 e 7 segundos. Todos os testes foram realizados sob as mesmas circunstâncias, tornando essa alteração de tempo uma incógnita para o grupo.

(**)-Observação 2: O teste do arquivo “entrada40.txt” foi realizado entre 23:47h de sábado, 05 de novembro de 2022, e 12:26h de domingo 06 de novembro de 2022. O programa ainda apresentava como resultado, de maneira extremamente lenta, novos arranjos, contudo o programa precisou ser interrompido para que novos testes nos demais arquivos fossem realizados para assim obtermos resultados mais precisos quanto ao tempo de execução das outras entradas.

Observação 3: Algumas falhas foram detectadas na compatibilidade do arranjo com a linha da matriz, mas o grupo não conseguiu corrigir estes resultados.

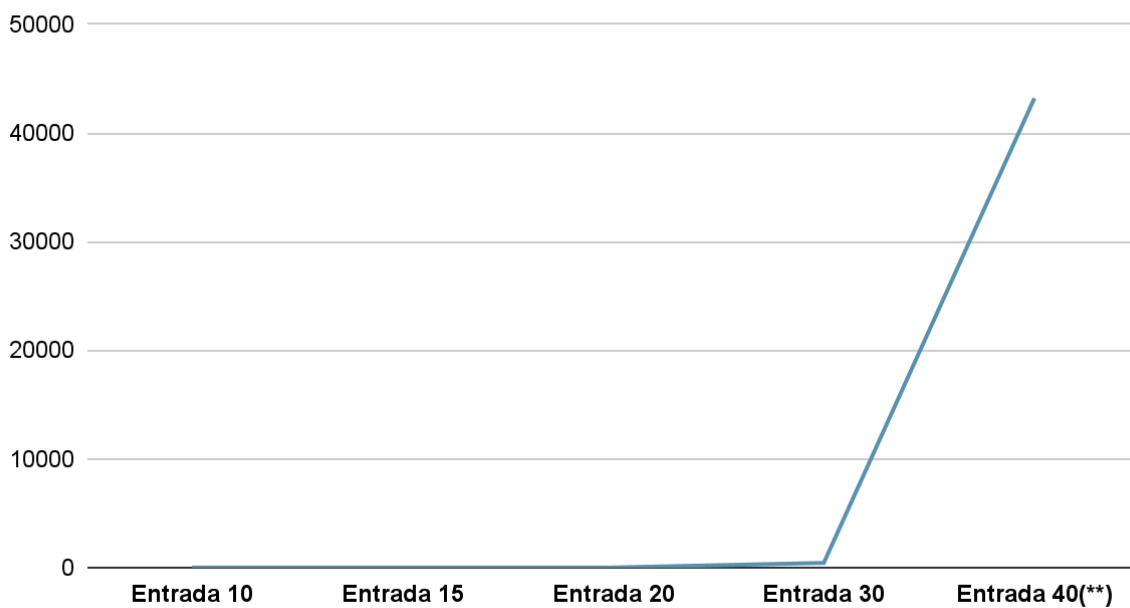
Observação 4: Todos os testes foram realizados em um notebook Dell Inspiron 15 - 5566 com processador Intel Core i5, com 8GB de memória RAM e com Sistema Operacional Linux Ubuntu. A seguir estão algumas informações sobre este computador.

Processador	Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz
Arquitetura	x86_64
CPU MHz máx.	3100,0000
CPU MHz mín.	400,0000
L1d:	64 KB (2 instances)
L1i:	64 KB (2 instances)
L2:	512 KB (2 instances)
L3:	3 MB (1 instance)

Imagem 6: Algumas informações sobre o notebook

Gráfico 1: Tempo de execução dos arquivos de entrada em segundos

Tempo de execução do arquivo de entrada em segundos



7. Referências

Para a realização deste trabalho utilizados alguns materiais que serão listados abaixo:

1 - Livro base da disciplina:

Projeto de Algoritmos com implementações em PASCAL e C - Nivio Ziviani

2 - Para gerar os arranjos:

Gerando Permutações-r com Repetição em C - disponível em

<https://daemoniolabs.wordpress.com/2011/02/11/gerando-permutacoes-r-com-repeticao-em-c/> - Acessado em 31/10/2022