

Teoria de Grafos

Kaway Henrique da Rocha Marinho

26 Agosto 2024

1 Questão 1

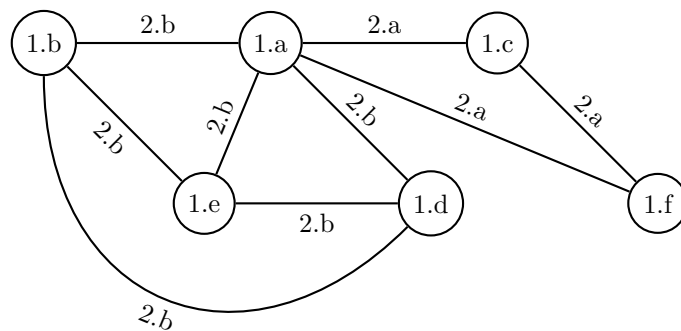
1. Atores Seleccionados

- (a) Matt Damon
- (b) Zazie Beetz
- (c) Wagner Moura
- (d) Brad Pitt
- (e) Morena Baccarin
- (f) Alice Braga

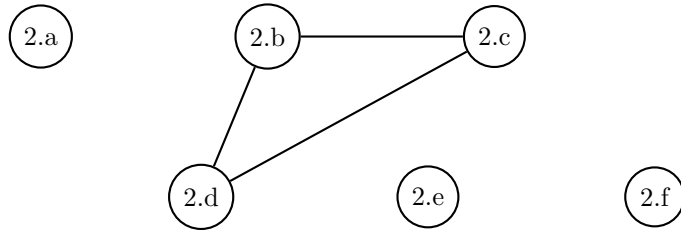
2. Filmes Seleccionados

- (a) Elysium - Neill Blomkamp
- (b) Deadpool 2 - David Leitch
- (c) Trem-Bala - David Leitch
- (d) John Wick - David Leitch
- (e) Cidade de Deus - Fernando Meirelles
- (f) Nove Dias - Edson Oda

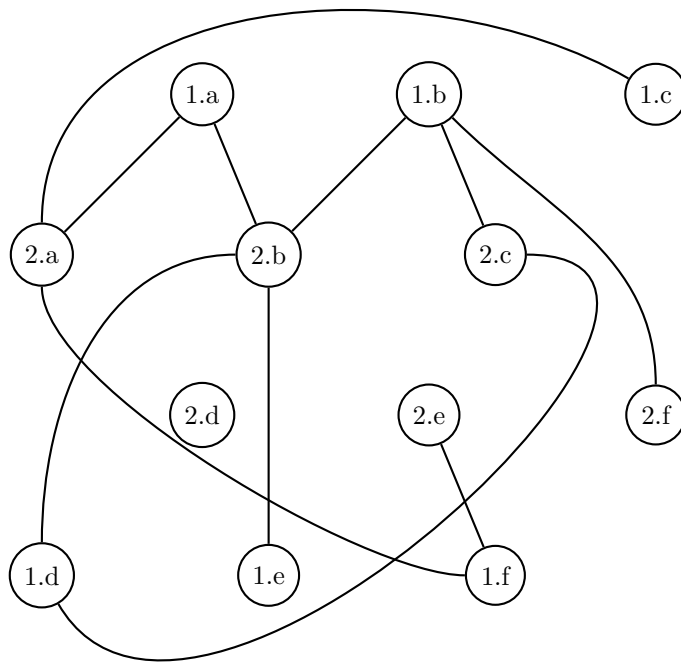
1.1 Objetos são atores



1.2 Objetos são filmes



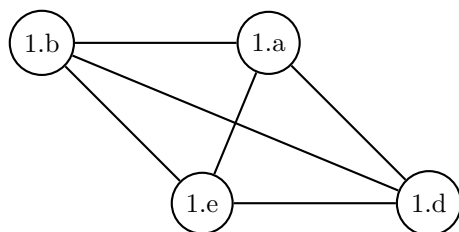
1.3 Objetos são filmes e atores



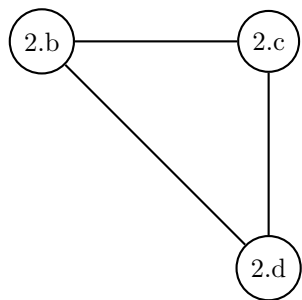
2 Questão 2

2.1 Maior Clique do Grafo

2.1.1 Grafo 1.1

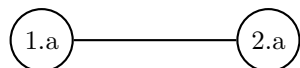


2.1.2 Grafo 1.2



2.1.3 Grafo 1.3

Esse grafo possui apenas cliques triviais de apenas uma aresta, podendo ser exemplificado por:



2.2 Diâmetro do Grafo

2.2.1 Grafo 1.1

O diâmetro desse grafo é 2.

2.2.2 Grafo 1.2

O diâmetro desse grafo é 1.

2.2.3 Grafo 1.3

O diâmetro desse grafo é 5.

2.3 Existência de Ciclo Hamiltoniano

2.3.1 Grafo 1.1

Não existe um ciclo Hamiltoniano nesse grafo, verificado por inspeção visual.

2.3.2 Grafo 1.2

Não existe um ciclo Hamiltoniano nesse grafo, verificado por inspeção visual.

2.3.3 Grafo 1.3

Não existe um ciclo Hamiltoniano nesse grafo, verificado por inspeção visual.

2.4 Existência de Ciclo Euleriano

2.4.1 Grafo 1.1

Não existe um ciclo Euleriano nesse grafo, verificado por inspeção visual.

2.4.2 Grafo 1.2

Não existe um ciclo Euleriano nesse grafo, verificado por inspeção visual.

2.4.3 Grafo 1.3

Não existe um ciclo Euleriano nesse grafo, verificado por inspeção visual.

2.5 Bipartição do Grafo

2.5.1 Grafo 1.1

Não existem conjuntos V_1 e V_2 tais que a intersecção entre eles seja o conjunto vazio.

2.5.2 Grafo 1.2

Não existem conjuntos V_1 e V_2 tais que a intersecção entre eles seja o conjunto vazio.

2.5.3 Grafo 1.3

Por definição este grafo é bipartido, já que relaciona dois tipos diferentes de dados (atores e filmes).

3 Questão 3

Os atores escolhidos foram os mesmos presentes na Questão 1:

1. Wagner Moura - 2
2. Morena Baccarin - 2
3. Alice Braga - 2

O maior número (finito) que encontrei foi para o ator Leandro Mazzini, principal de uma série B brasileira. Passando por filmes como "Os homens são de Marte e é pra lá que eu vou" o número de Bacon do ator é 4.

4 Questão 4

Meu número de Erdős é 4, seguindo o caminho:

Marinho, K. \leftarrow Figueiredo, Daniel Ratton \leftarrow Reed, Bruce A. \leftarrow Alon, Noga
 \leftarrow Erdős, Paul

5 Questão 5

5.1 Caminho entre UFRJ e MIT

1. www.ufrj.br
2. <https://internacional.ufrj.br/>
3. <https://internacional.ufrj.br/pesquisaacordos/>
4. <https://www.nyu.edu/>
5. <https://x.com/nyuniversity>
6. https://x.com/i/connect_people?user_id=28421825 (Página de sugestões de follow do X)
7. <https://x.com/MIT>
8. <http://socialmediahub.mit.edu/>
9. <https://web.mit.edu/>

O comprimento total do caminho é 8.

5.2 Caminho alternativo entre UFRJ e MIT

1. <https://ufrj.br/>
2. <https://acessibilidade.ufrj.br/>
3. <https://www.gov.br/pt-br>
4. <https://creativecommons.org/licenses/by-nd/3.0/deed.en>
5. <https://creativecommons.org/share-your-work/cclicenses/>
6. <https://creativecommons.org/faq/>
7. <https://ocw.mit.edu/>
8. <https://openlearning.mit.edu/>
9. <https://openlearning.mit.edu/accessibility>
10. <https://accessibility.mit.edu/>
11. <https://web.mit.edu/>

O comprimento total do caminho é 10.

5.3 Como encontrar o menor caminho possível

Uma forma simples que acredito cumprir o trabalho, apesar de ser extremamente não eficiente, é "inundar" todos os possíveis caminhos entre dois vértices presentes no grafo e após isso selecionar o que possuir menor valor. Essa abordagem no entanto poderia ser muito demorada, e sem uma inteligência maior para o algoritmo ele acabaria repetindo muitos caminhos ou não chegando a lugar algum em algumas ocasiões

6 Questão 6

1. O programa imprimirá $n!$ permutações.
2. Segundo a Wikipedia a estimativa de átomos no universo é de 10^{80} . Conforme o item 6.1, podemos testar valores utilizando uma calculadora online, onde chegamos a uma entrada do valor 59.
3. Os dados gerados conforme a tabela 1 geram o seguinte gráfico:

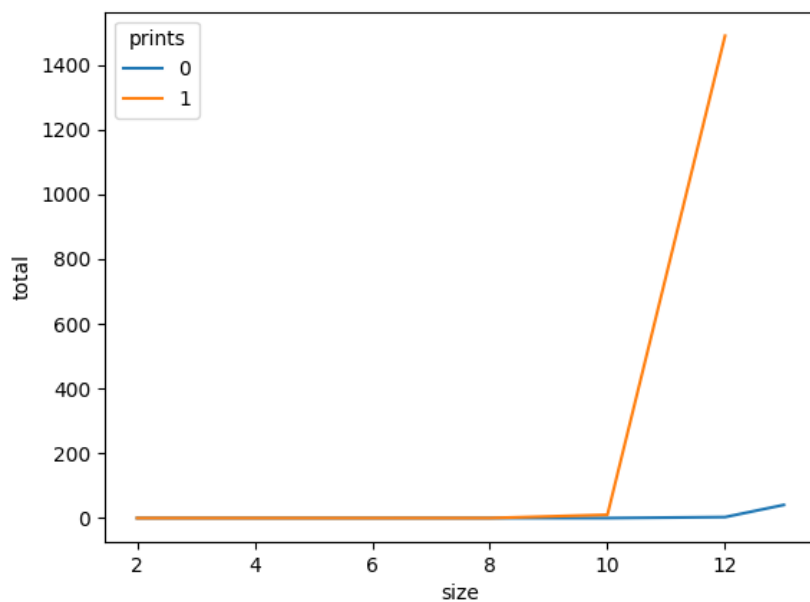


Figure 1: Gráfico de performance do programa

É possível notar um "spike" no tempo levado para a computação quando ocorre o print da permutação, isso reflete não somente a natureza fatorial do problema mas também os custos de entregar aquela saída para o STD-OUT. É possível notar que isso é um fator pois os valores para quando não ocorre o print da permutação são consideravelmente menores, o que me permitiu também computar mais uma iteração (13!).

size	user	system	total	prints
2	0	0	0	1
4	0	0	0	1
6	0	0	0.01	1
8	0.03	0.05	0.1	1
10	3.27	3.93	10.25	1
12	474.51	527.24	1489.81	1
2	0	0	0	0
4	0	0	0	0
6	0	0	0	0
8	0	0	0	0
10	0.04	0	0.04	0
12	3.12	0	3.13	0
13	40.68	0	40.69	0

Table 1: Valores gerados pelo programa

Código utilizado para a computação das permutações:

```
pub fn permute(left: usize, right: usize,
list: &mut Vec<Option<i32>>) {
    if left == right {
        print_list(list);
    } else {
        for i in left..right + 1 {
            swap(left, i, list);
            permute(left + 1, right, list);
            swap(left, i, list);
        }
    }
}
```

Sendo a função swap definida como:

```
fn swap(left: usize, right: usize,
list: &mut Vec<Option<i32>>) {
    if left == right {
        return;
    }
    let temp1 = list[left].take().unwrap();
    let temp2 = list[right].take().unwrap();
    list[left] = Some(temp2);
    list[right] = Some(temp1);
}
```