Trabalho Prático - Matemática Discreta - 2022/1
Aluna: Aline Cristina Pinto **Matrícula:** 2020031412

O problema da celebridade

→ Entendendo o problema

*Num conjunto S de n pessoas, **apenas uma pessoa** é conhecida por todos. Essa pessoa **pode estar presente** na festa, se sim, ela não conhece ninguém na festa. Só podemos fazer perguntas como "**A conhece B?**". Queremos saber se existe uma celebridade em S.*

Analogamente, o problema pode ser formalizado da seguinte forma:

Dado um conjunto de n pessoas e a matriz associada M, encontrar (se existir) uma celebridade no conjunto. Isto é, determinar um k tal que todos os elementos da coluna k (exceto M[k, k]) sejam 1s e todos os elementos da linha k sejam 0s.

```
M = { {0, 0, 1, 0},   Existe uma celebridade - posição i=2, j=2!
      {0, 0, 1, 0},
      {0, 0, 0, 0},
      {0, 0, 1, 0} }
```

```
M = { {0, 0, 1, 0},   Não existe uma celebridade!
      {0, 0, 1, 0},
      {0, 1, 0, 0},
      {0, 0, 1, 0} }
```

→ Soluções:

Método 1 - Grafo - $O(n^2)$:

Nessa abordagem, imaginamos o problema como um grafo, no qual:
- As pessoas são vértices
- Se A conhece B, então existe uma aresta orientada de A para B

Caso exista uma celebridade, então haverá um vértice no qual chegam n-1 arestas, mas não saem nenhuma aresta.
Teremos então dois arrays: indegree (número de arestas que chegam no vértice) e outdegree (número de arestas que saem do vértice) e vamos **construir todas as arestas possíveis para cada par possível [i, j] (A, B)**.

Algoritmo hipotético:
- São criados e inicializados com 0 dois vetores: indegree e outdegree.
- Com dois for aninhados, os vetores são percorridos de 0 a n
    - Se i conhece j, o vetor outdegree de i e o indegree de j são incrementados
    - Se j conhece i, o vetor outdegree de j e o indegree de i são incrementados

- Com um novo for de 0 a n, os dois vetores são percorridos de forma a encontrar um índice no qual o vetor outdegree seja 0 e vetor indegree seja n-1. Caso isso não seja possível, então não há uma celebridade no conjunto.

Complexidade:
Nesse caso, como para cada elemento estamos verificando todos os outros elementos (os dois for aninhados), **o custo dessa solução seria de 2n(n-1), ou seja, complexidade $O(n^2)$.**

Método 2 - Eliminação - $O(n)$:

Nessa técnica, é observado que:
- Se A conhece B, A não pode ser a celebridade e é descartado, mas B pode.
- Se A não conhece B, então B não pode ser a celebridade e é descartado, mas A pode ser.

Esse processo é repetido até restar apenas uma pessoa (possível celebridade). Se n-1 pessoas conhecem essa pessoa e a mesma conhece 0 pessoas, então a celebridade foi encontrada, caso contrário não há celebridade.

Algoritmo hipotético:
- É criada uma pilha com os id's das pessoas
- Com um while (enquanto há mais de um id na pilha) os dois primeiros elementos da pilha (representando A e B) são removidos (pop):
    - Se A conhece B, então B é devolvido para a pilha (push)
    - Mas se A não conhece B, então A é devolvido para a pilha (push)
- Ao fim, o elemento que restou na pilha é atribuído como a possível celebridade e com o auxílio de um for de 0 a n-1, é contado o número de pessoas que conhecem essa possível celebridade e o número que a mesma conhece:
    - Se n-1 pessoas conhecem a possível celebridade e a mesma conhece 0 pessoas, então a celebridade foi encontrada, caso contrário, o conjunto não possui celebridade.

Complexidade:
Para determinar se uma pessoa não conhece ninguém e que todos a conhecem, são necessárias $2(n - 1)$ perguntas.
Como observado acima, é possível descobrir e remover quem não é a celebridade com apenas **uma pergunta**. E, ao restar apenas um candidato (do qual teremos que verificar se é conhecido por todos e vice-versa), temos que fazer **2(n - 1) perguntas**.
Então o total de perguntas (comparações) foi $3(n - 1)$, ou seja, complexidade $O(n)$.

Método 3 - Recursão - $O(n)$:

Essa estratégia, utiliza a mesma lógica do método de eliminação, mas com o auxílio da recursão:

Algoritmo hipotético:

- É criada uma função recursiva que recebe um inteiro n (número de pessoas)
- É verificado o caso base: se n é 0, então retorna -1 (não há celebridades possíveis pois há 0 pessoas)
- A função recursiva é chamada para os primeiros n-1 elementos e retorna o id da celebridade em potencial:
    - Se o id recebido é -1, então não há celebridades e n-1 é retornado
    - Senão, se o id recebido conhece n-1, então o id não é a celebridade e n-1 é retornado
    - Senão, se n-1 conhece o id recebido, então esse id pode ser a possível celebridade e é retornado
    - Senão -1 é retornado
- Ao fim, o id retornado a função principal é atribuído como a possível celebridade
    - Se o id é -1, então não há celebridade;
    - Se o id é diferente de -1, com o auxílio de um for de 0 a n-1, é contado o número de pessoas que conhecem essa possível celebridade e o número que a mesma conhece:
        - Se n-1 pessoas conhecem a possível celebridade e a mesma conhece 0 pessoas, então a celebridade foi encontrada, caso contrário, o conjunto não possui celebridade.

Complexidade:

Neste caso, a **função recursiva é chamada n vezes**, logo, **a complexidade é $O(n)$.**

→ Referências:

https://www.geeksforgeeks.org/the-celebrity-problem/?ref=lbp
https://www.ic.unicamp.br/~rezende/ensino/mo417/2010s2/Slides/Aula7a.pdf
https://sheilaalmeida.pro.br/aulas/apa/aula07-ProjetoPorInducao.pdf