

Problema de Josephus

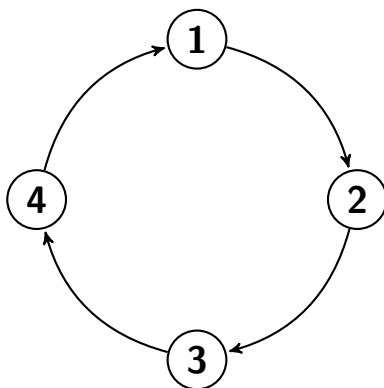
Conta-se uma história sobre o fim de uma guerra hebraica, na qual o historiador Josephus e mais 40 homens foram cercados pelo exército romano. Eles decidiram não se entregar com vida, mas sua religião condena o suicídio. Por isso, fizeram um pacto no qual os 41 homens fariam um círculo e, começando pelo homem na primeira posição, um homem de cada vez mataria aquele imediatamente à sua esquerda, sendo que o próximo homem a matar é o que estava a esquerda do último a morrer. Nesse contexto, Josephus precisou encontrar em qual lugar no círculo deveria se posicionar para que fosse o último homem que restasse. Essa história dá origem ao **Problema de Josephus** que será o exercício a ser resolvido nesse trabalho. O vídeo “The Josephus Problem - Numberphile” é bastante interessante e explora mais a fundo este problema.

Para resolver o problema, utilizaremos o conceito de uma lista ligada circular, por isso será necessário implementar algumas funções que auxiliem a criação e manipulação de listas encadeadas, além da resolução do problema em si.

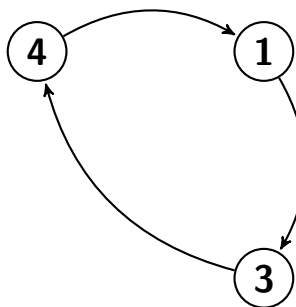
1 Resolvendo o problema

Suponha que n pessoas estão dispostas em um círculo e que são numeradas de 1 a n no sentido horário. Comece pela pessoa de número 1 e elimine a $(m + 1)$ -ésima pessoa. Em seguida, contando a partir da pessoa p que sucede a última eliminada, elimine a pessoa que está m posições depois de p . Repita este processo enquanto o círculo tiver duas ou mais pessoas. Note que, como estamos falando de um círculo, ao andar m posições a partir de uma pessoa, podemos acabar chegando em pessoas que estão antes dela, ou até mesmo a ela própria. Como a pessoa nunca pode matar a si própria, neste caso ela deve matar quem estiver imediatamente à frente dela na lista.

Segue um exemplo da resolução do problema com $n = 4$ e $m = 1$:



A primeira pessoa a ser eliminada é a que se encontra na posição 2.



A próxima pessoa eliminada se encontra na posição 4, já que 3 sucede o último eliminado e $m = 1$ posições depois de 3 está o 4.



A última pessoa eliminada se encontra na posição 3 e a sobrevivente é a que se encontra na posição 1.



Sua implementação deve necessariamente conter uma função chamada *solveJosephus* que recebe dois parâmetros, o número de pessoas n e o tamanho do passo m e retornar a posição do sobrevivente. A função pode receber mais parâmetros se for conveniente para sua implementação.

Além disso, a **solução deve estar baseada em uma estrutura encadeada**.

2 Exemplo de caso de testes

Cada caso de teste é composto por um ou mais instâncias problemas.

Na primeira linha há o número de casos de teste. Então, em seguida, há o número de pessoas n e o tamanho do passo m para cada instância do problema.

A saída esperada deve seguir o formato apresentado. Onde o resultado indica a posição do sobrevivente.

Entrada:

```
1 3
2 10
3 1
4 10
5 2
6 10
7 3
```

Saída esperada:

```
1 Usando n=10, m=1, resultado=5
2 Usando n=10, m=2, resultado=10
3 Usando n=10, m=3, resultado=6
```

3 Referências interessantes

Josephus Problem - Wikipedia: https://en.wikipedia.org/wiki/Josephus_problem

The Josephus Problem - Numberphile (vídeo) <https://www.youtube.com/watch?v=uCsD3ZGzMgE>

Site Projeto de Algoritmos: <https://www.ime.usp.br/~pf/algoritmos/>