# Exemplos de Sala de Aula
# Semana 2

1. Suponha que existem $n$ pessoas em um grupo, cada pessoa está ciente de um escândalo que ninguém mais no grupo sabe sobre. Essas pessoas conversam por telefone; quando duas pessoas no grupo conversam, elas compartilham informações sobre todos os escândalos que cada um conhece. Para por exemplo, na primeira chamada, duas pessoas compartilham informações, então ao final da ligação, cada uma dessas pessoas sabe sobre dois escândalos. O problema da fofoca pede $G(n)$, o mínimo número de chamadas telefônicas necessárias para todas as n pessoas para aprenda sobre todos os escândalos.

   a) Mostre $G(1) = 0$

   b) Mostre $G(2) = 1$

   c) Mostre $G(3) = 3$

   d) Mostre $G(4) = 4$

   e) Mostre $G(5) = 6$

   f) Proponha um definição recursiva para $G(n)$

**Algorithm 1** Gossip(n)

**Require:** $n$ representando o número pessoas
**Ensure:** Sequência de ligações realizadas para compartilhar todas as fofocas

```
if n == 1 then
    Nenhuma ligação será realizada.
else if n == 2 then
    Pessoa 1 liga para pessoa 2
else if n == 3 then
    Pessoa 1 liga para pessoa 2
    Pessoa 2 liga para pessoa 3
    Pessoa 3 liga para pessoa 1
else if n == 4 then
    Pessoa 1 liga para pessoa 2
    Pessoa 3 liga para pessoa 4
    Pessoa 1 liga para pessoa 4
    Pessoa 2 liga para pessoa 3
else
    Pessoa n liga para pessoa 1
    Gossip(n-1)
    Pessoa 1 liga para pessoa n
end if
```

2. Um convidado em uma festa é uma celebridade se essa pessoa for conhecida por todos os outros convidados, mas não conhece nenhum deles. Existe no máximo uma celebridade em uma festa, pois se houvesse duas, eles se conheceriam.. Uma festa particular pode não ter celebridade. Sua tarefa é encontrar a celebridade, se ela existe, em uma festa, fazendo apenas um tipo de pergunta - perguntando a um convidado se ele conhece um segundo convidado. Todos devem responder às suas perguntas com sinceridade. Ou seja, se Alice e Bob são duas pessoas na festa, você pode perguntar a Alice se ela conhece Bob; ela deve responder corretamente.

   Seja $G(n)$ o número de perguntas utilizada para encontrar uma celebridade em uma festa com n pessoas.

   a) Calcule $G(1)$, $G(2)$, $G(3)$

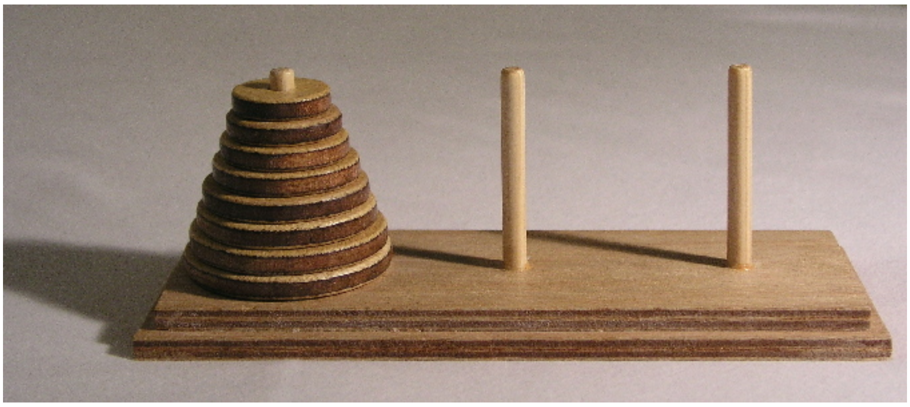
   b) Proponha uma definição recursiva para $G(n)$. [Dica: primeiro faça uma pergunta para eliminar uma pessoa como celebridade. Em seguida, identifique uma celebridade em potencial. Por fim, faça mais duas perguntas para determinar se a celebridade em potencial é na verdade uma celebridade.]

**Algorithm 2** celebridade(S)

**Require:** $S$ representando um conjunto de pessoas em uma festa.
**Ensure:** Sequência de perguntas para encontrar a celebridade da festa

```
if |S| == 1 then
    return v ∈ S
else
    i, j ∈ S com i ≠ j
    if conhece(i, j) then
        S ← S \ {i}
        k ← celebridade(S)
        if conhece(i, k) and ¬conhece(k, i) then
            return k
        end if
    else
        S ← S \ {j}
        k ← celebridade(S)
        if conhece(j, k) and ¬conhece(k, j) then
            return k
        end if
    end if
end if
```

3. O quebra-cabeça da Torre de Hanói consiste em três pinos verticais e vários discos de vários tamanhos. Cada disco tem um buraco no centro para os pinos passarem.



As regras do quebra-cabeça são as seguintes:

- O quebra-cabeça começa com todos os discos colocados em um dos pinos. Eles são colocados em ordem do maior para o menor, de baixo para cima.
- O objetivo do quebra-cabeça é mover todos os discos para outro pino.
- Apenas um disco pode ser movido por vez, e os discos são sempre colocados em pinos.
- Os discos só podem ser movidos para um pino vazio ou para um disco maior.

**Algorithm 3** hanoi(n, origem, trabalho, destino)

**Require:** $n$ representando o número de discos colocados no pino inicial, $origem$ representando o rótulo do pino inicial, $trabalho$ representando o rótulo do pino de trabalho e $destino$ representando o rótulo do pino de destino.
**Ensure:** Sequência de transferência de discos entre os pinos

```
if n == 1 then
    Mova disco 1 do pino de origem para o pino de destino
else
    hanoi(n-1, origem, destino, trabalho)
    Mova o disco n do pino de origem para o pino de destino.
    hanoi(n-1, trabalho, origem, destino)
end if
```

4. Reza a lenda que Flávio Josephus participou de uma rebelião contra o poderoso Império Romano. Seu grupo foi encurralado pelos romanos em uma caverna. O grupo preferindo o suícidio à captura. O grupo resolveu formar um círculo e percorren no sentido horário matar cada segunda pessoa até não sobrar ninguém. Conta a lenda que Flávio Josephus calculou rapidamente a posição que ele e seu amigo deveriam ocupar para que eles a sobrevivência fosse garantida.

   Na nossa versão, queremos saber qual é a posição que deve ser ocupada por Flávio Josephus para que ele seja o último eliminado.

   Para desenvolver a relação de recorrência, vamos utilizar uma tabela com os sobreviven Considere o caso de $n = 10$:

| Posição | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Pessoa | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

   Na primeira etapa, podemos remover todas as pessoas nas posições pares:

| Posição | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Pessoa | 1 | 3 | 5 | 7 | 9 |

   Note agora que temos o mesmo problema com o número de pessoas reduzido pela metade e podemos fazer um mapeamento entre a posição da pessoa e o seu identificados, basta multiplicar por 2 e subtrair uma unidade. Logo, temos a seguinte relação:

$$J(2n) = 2J(n) - 1$$

   Agora, vamos analisar o caso em que n é ímpar. Considere $n = 9$:

| Posição | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Pessoa | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

Na primeira etapa, podemos remover todas as pessoas nas posições pares e a primeira posição:

| Posição | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Pessoa | 3 | 5 | 7 | 9 |

Novamente, reduzimos o problema original pela metade e podemos fazer um mapeamento entre a posição da pessoa e o valor do seu identificador, basta multiplicar por 2 e somar uma unidade. Logo, temos a seguinte relação de recorrência:

$$J(2n+1) = 2J(n) + 1$$

A relação de recorrência completa é a seguinte:

$$\begin{array}{lcl} J(1) & = & 1 \\ J(2n) & = & 2J(n) - 1 \\ J(2n+1) & = & 2J(n) + 1 \end{array}$$

5. O número de maneiras de selecionar $k$ pessoas em um grupo de $n$ pessoas pode ser definido de maneira recursiva escolhendo uma pessoa qualquer $x$ do grupo e considerando dois casos:

   - Se x é selecionado, precisamos selecionar $k-1$ pessoas de um grupo com $n-1$ pessoas.
   - Se x não é selecionado, precisamos selecionar $k$ pessoas de um grupo com $n-1$ pessoas.

   Seja $\binom{n}{k}$ representando o número de maneiras de escolher $k$ pessoas em um grupo de $n$ pessoas.

   A relação recorrente que define $\binom{n}{k}$ é a seguinte:

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}, \text{ para } 0 < k < n$$
$$\binom{n}{0} = 1$$
$$\binom{n}{n} = 1$$

   Preencha a seguinte tabela usando a relação acima:

| $n/k$ | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 1 | X | X | X | X |
| 1 | 1 | 1 | X | X | X |
| 2 | 1 | 2 | 1 | X | X |
| 3 | 1 | 3 | 3 | 1 | X |
| 4 | 1 | 4 | 6 | 4 | 1 |