<u>Backtracking</u>: um método sistemático para encontrar todas as configurações de um espaço de busca.

Cada passo de um alg. backtracking, começa com uma solução parcial e tentamos estendê-la adicionando um novo elemento à solução. Depois de estender a solução devemos verificar se temos solução total. Se sim, processamos tal solução. Se não, devemos verificar se a solução parcial ~~é~~ pode ser estendida. Se sim, continue. Se não, remova o último elemento inserido e tente estender tal solução parcial de outra forma.

Problema: Construindo todos os subconjuntos de ②

$|S| = n$. Sabemos que S possui 2^n subconjuntos.

$S = \{1, 2, 3\}$

$\mathcal{P}(S) = \{ \{1,2,3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1\}, \{2\}, \{3\}, \{\} \}$.

$L^1 = [\overset{1}{\underline{V}}, \overset{2}{\underline{V}}, \overset{3}{\underline{V}}]$

$L^2 = [\overset{1}{\underline{V}}, \overset{2}{\underline{V}}, \overset{3}{\underline{F}}]$

$L^3 = [\overset{1}{\underline{V}}, \overset{2}{\underline{F}}, \overset{3}{\underline{F}}]$

$L = [\overset{1}{V}, V, V]$
$= [V, V, F]$
$= [V, F, V]$
$= [V, F, F]$
$= [F, V, V]$
$= [F, V, F]$

A = [1 ... n] A possui n posições e A é booleano.

K → representa o tamanho da solução atual

```
Subconjuntos (A, K, n)
    se K = n
        para i ← 1 até n faça
            escreva "i" se A[i] = V
    senão A[K+1] ← V
        Subconjuntos (A, K+1, n)
        A[K+1] ← F
        Subconjuntos (A, K+1, n)
```

Chamada:

```
Subconjuntos (A, 0, n)
```

Problema: Construir todas as permutações de $1, \dots, n$.

Ex.: $n = 3 \xRightarrow{\text{tot.}} 3! = 6$.

1, 2, 3
1, 3, 2
2, 1, 3
2, 3, 1
3, 1, 2
3, 2, 1

$A = [0, 0, 0]$

$A = [1, 0, 0]$

$A = [1, 2, 0]$.

$\boxed{A = [1, 2, 3]}$.

$A = [1, 0, 2]$

$\boxed{A = [1, 3, 2]}$.

$A = [0, 1, 0]$

$A = [2, 1, 0]$.

$\boxed{A = [2, 1, 3]}$.

$A = [0, 1, 2]$

$\boxed{A = [3, 1, 2]}$.

$A = [0, 0, 1]$

$A = [2, 0, 1]$

$\boxed{A = [2, 3, 1]}$.

$A = [0, 2, 1]$

$\boxed{A = [3, 2, 1]}$.

Recebe: A é lista com n elementos (inicializ. A[i] = 0 $\forall$ i). tamanho(A)

K é o tamanho da solução atual.

Devolve: Todas as permutações.

Permutações (A, K)

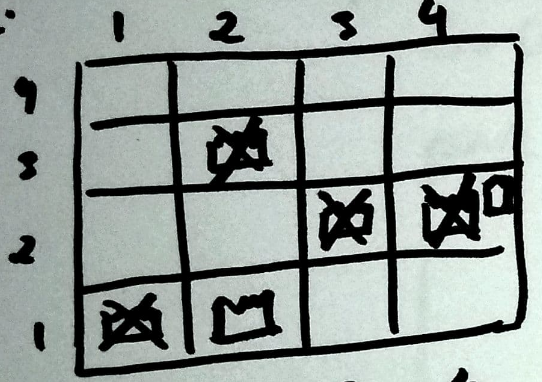
```
se K = tamanho(A)
    escreve (A)
senão
    para cada i ← 1 até tamanho (A) faça
        se A[i] = 0
            A[i] = K+1
            Permutações (A, K+1)
            A[i] = 0
```

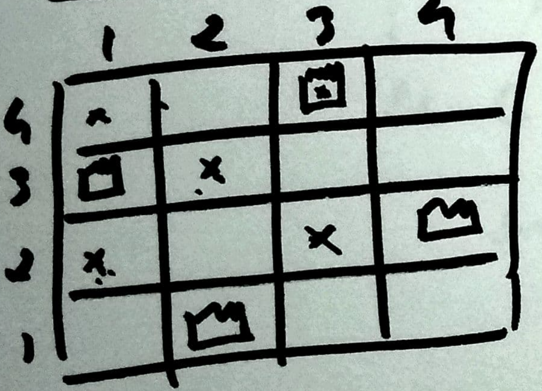
Chamada: A=[0, 0, ..., 0]

Permutações (A, 0)

Problema das 8 rainhas: De quantas formas podemos colocar 8 rainhas em um tabuleiro 8x8 de maneira que nenhuma rainha ataque qualquer outra rainha?

Ex.:





$$A = \begin{array}{cccc} 1 & 2 & 3 & 4 \\ \end{array}$$

$$A = [\begin{array}{cccc} \cancel{1} & \cancel{3} & \cancel{2} & 3 \\ 2 & \cancel{4} & \cancel{1} & \\ & 4 & & \end{array}]$$

$((3),(1))$, $((3),(2))$

$|2-3| = |1-2|$

Seja A uma lista de inteiros. Colocaremos em $A[l]$ algum valor no intervalo 1 até 8. Se $A[l] = c$, então temos uma rainha na linha l e coluna c do tabuleiro.

8_Rainhas (A, K) ▷ n_soluções → variável global.

linha coluna
A[K+1, c].
A[i, A[i]]

```
se K = 8
    então n_soluções = n_soluções + 1
    senão para c ← 1 até 8 faça
            c_candidato = V
            para i ← 1 até K faça
                se A[i] = c então c_candidato = F
                se |(K+1) - i| = |A[i] - c| então c_candidato = F
            se c_candidato = V
                então A[K+1] ← c
                      8_Rainhas (A, K+1)
```

Chamada:

8_Rainhas (A, 0) → vazio