Técnica de Resolução de Problemas: Força Bruta

Prof. Daniel Kikuti

Universidade Estadual de Maringá

Método força bruta (busca exaustiva)

O que é?

É uma técnica genérica para resolução de problemas que consiste em sistematicamente enumerar todos os possíveis candidatos a solução e verificar se cada candidato satisfaz o enunciado do problema (**gerar e testar**).

Exemplos

- Encontrar os divisores de um número natural n.
- Colocar oito rainhas em um tabuleiro de xadrez 8×8 sem que uma rainha ataque outra.

Visão geral do algoritmo força bruta

Dada uma instância P de um problema a ser resolvido, o algoritmo a seguir descreve o funcionamento do método:

$\textsc{Brute-Force}(P)$

```
c ← First(P)
while c ≠ nil do
    if Valid(P, c) then  Output(P, c)
    c ← Next(P, c)
```

Significado das funções

1. $\textsc{First}(P)$ gera o primeiro candidato para P.
2. $\textsc{Next}(P, c)$ gera o próximo candidato para P após o atual c.
3. $\textsc{Valid}(P, c)$ verifica se c é solução para P.
4. $\textsc{Output}(P, c)$ use a solução c de P na aplicação.

Problema 1

Enumerar todos os subconjuntos

Dado um conjunto S com n elementos, listar todos os subconjuntos que podem ser gerados a partir de elementos de S.

Exemplo

$S = \{a, b, c, d\}$, $n = 4$.

$$
\begin{array}{lllll}
\{\} & \{a\} & \{a, b\} & \{a, b, c\} & \{a, b, c, d\} \\
 & \{b\} & \{a, c\} & \{a, b, d\} & \\
 & \{c\} & \{a, d\} & \{a, c, d\} & \\
 & \{d\} & \{b, c\} & \{b, c, d\} & \\
 & & \{b, d\} & & \\
 & & \{c, d\} & &
\end{array}
$$

Pergunta: Para um conjunto S de tamanho n, é possível gerar quantos subconjuntos distintos?

Problema 1

Enumerar todos os subconjuntos

Dado um conjunto S com n elementos, listar todos os subconjuntos que podem ser gerados a partir de elementos de S.

Exemplo

$S = \{a, b, c, d\}$, $n = 4$.

$$
\begin{array}{ccccc}
\{\} & \{a\} & \{a, b\} & \{a, b, c\} & \{a, b, c, d\} \\
 & \{b\} & \{a, c\} & \{a, b, d\} & \\
 & \{c\} & \{a, d\} & \{a, c, d\} & \\
 & \{d\} & \{b, c\} & \{b, c, d\} & \\
 & & \{b, d\} & & \\
 & & \{c, d\} & &
\end{array}
$$

Pergunta: Para um conjunto S de tamanho n, é possível gerar quantos subconjuntos distintos? 2^n.

Representação binária para geração de subconjuntos

a	b	c	d	Conjunto
0	0	0	0	$\{\}$
0	0	0	1	$\{d\}$
0	0	1	0	$\{c\}$
0	0	1	1	$\{c, d\}$
0	1	0	0	$\{b\}$
0	1	0	1	$\{b, d\}$
0	1	1	0	$\{b, c\}$
0	1	1	1	$\{b, c, d\}$
1	0	0	0	$\{a\}$
1	0	0	1	$\{a, d\}$
1	0	1	0	$\{a, c\}$
1	0	1	1	$\{a, c, d\}$
1	1	0	0	$\{a, b\}$
1	1	0	1	$\{a, b, d\}$
1	1	1	0	$\{a, b, c\}$
1	1	1	1	$\{a, b, c, d\}$

Dado o conjunto S e número de elementos n, o algoritmo a seguir imprime todos subconjuntos.

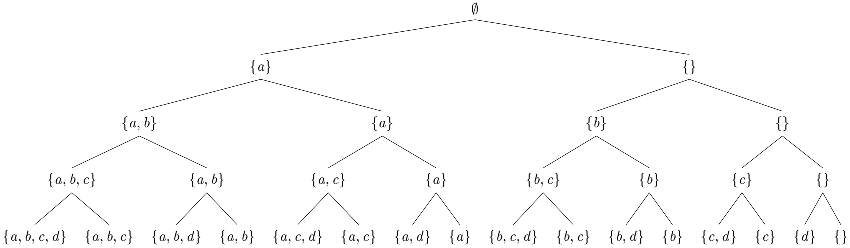
$\textsc{Subconjuntos}(S, n)$

1 **for** $num \leftarrow 0$ **to** $2^n - 1$ **do**
2     $C \leftarrow \emptyset$
3   **for** $i \leftarrow 0$ **to** $n - 1$ **do**
4   **if** $\textsc{Bit}(num, i) = 1$ **then**
5             $C \leftarrow C \cup S[i]$
6   Imprima os elementos de C

A função $\textsc{Bit}$ verifica se o i-ésimo bit de num é igual a 1. Na linguagem C, poderíamos substituir a linha 4 por:

```
if (num & (1 << i))
```

Pensando no problema como uma Árvore de Decisão



Algoritmo recursivo

$\textsc{Subconjunto-Rec}(S, X, i, n)$

if $i > n$ **then**
  Imprima elementos de X
else
  $X \leftarrow X \cup S[i]$
  $\textsc{Subconjunto-Rec}(S, X, i + 1, n)$
  $X \leftarrow X \setminus S[i]$
  $\textsc{Subconjunto-Rec}(S, X, i + 1, n)$

Legenda

- X contém um conjunto (inicialmente vazio).
- i é o índice do i-ésimo elemento de S.
- Chamada inicial: $\textsc{Subconjunto-Rec}(S, X, 0, |S| - 1)$.

Exercício

Soma de subconjunto

Dado um conjunto de números inteiros S e um valor x, determine se existe um subconjunto de S cuja soma é igual a x. Exemplo: Para $S = \{3, 7, 5, 11, 1\}$ e $x = 9$ a resposta é sim ($\{3, 5, 1\}$). Para o mesmo conjunto S e $x = 2$ a resposta é não.

Implemente um algoritmo força bruta que resolve este problema. Considere uma instância de entrada inicial em que S possui 20 números. Teste seu algoritmo com instâncias maiores (sugestão 30, 35, 40 números).

Problema 2

Enumerar todos as permutações de um conjunto

Dado um conjunto S com n elementos, listar todas as permutações que podem ser geradas a partir dos elementos de S.

Exemplo

$S = \{a, b, c\}$, $n = 3$.

a, b, c

a, c, b

b, a, c

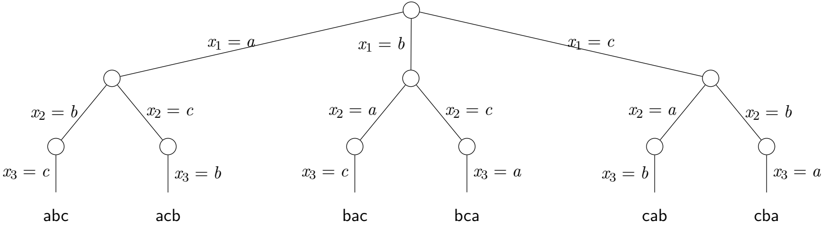
b, c, a

c, a, b

c, b, a

Pensando no problema como uma Árvore de Decisão

Considere $S = \{a, b, c\}$.



Algoritmo recursivo para permutação

$\textsc{Permuta}(S, i, n)$

if $i = n$ **then**
  Imprima elementos de S
else
  for $k \leftarrow i$ **to** n **do**
    $\textsc{Swap}(S[i], S[k])$
    $\textsc{Permuta}(S, i+1, n)$
    $\textsc{Swap}(S[i], S[k])$

Legenda

- Permutação feita no próprio conjunto S.
- i é o índice do i-ésimo elemento de S.
- Chamada inicial: $\textsc{Permuta}(S, 0, |S| - 1)$.

Exercício

Problema do Caixeiro Viajante

Dado um conjunto de cidades e as respectivas distâncias entre estas cidades, determine um percurso que passa por todas as cidades uma única vez e retorna ao ponto de origem com o menor custo possível.

Exemplo:

	a	b	c	d
a	0	1	1	5
b	1	0	7	1
c	1	7	0	1
d	5	1	1	0

O menor caminho para este conjunto de cidades seria $a \rightarrow b \rightarrow d \rightarrow c \rightarrow a$, com custo total 4.

Implemente um algoritmo força bruta que resolve este problema.