

Permutações e Combinações

Daniel Oliveira

Departamento de Informática - UFPR

Janeiro 2021



- Alguns padrões aparecem de forma recorrente na solução de problemas
 - Soluções algorítmicas exigem que tais padrões sejam gerados de forma metódica
- Permutações: ordem importa
 - Todas as formas de escrever ABC: "ABC", "ACB", "BAC", "BCA", ...
- Combinações: ordem não importa
 - Todas as sublistas de ABC: "ABC", "AB", "AC", "A", "BC", ...

Permutações de strings

Instância: s , onde s é uma string de tamanho n .

Resposta: Todas as permutações de s com tamanho n .

- Como abordar esse problema usando recursão?
- Para cada caracter da string s , coloco ele no início
 - 1 Gerar todas as permutações dos caracteres restantes (recursão com $n - 1$ caracteres)

Permutações - Algoritmo

```
permutar(pronto, restante)
```

```
Se restante é vazio  
  imprime(pronto)
```

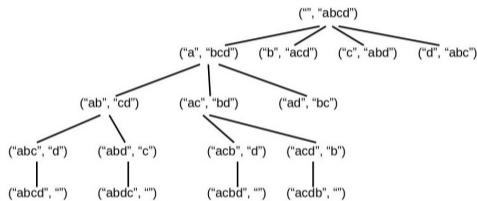
```
Senão
```

```
  Para todo caracter c em restante
```

```
    permutar(pronto + c, restante - c)
```

```
listar_permutações(string)
```

```
permutar("", string)
```



subconjuntos de strings

Instância: s , onde s é uma string de tamanho n .

Resposta: Todos os subconjuntos dos caracteres de s .

- Como abordar esse problema usando recursão?
- Considerando um caracter c da string s , coloco ele no conjunto ou não
 - 1 Gerar todos os subconjuntos dos caracteres restantes incluindo c (recursão)
 - 2 Gerar todos os subconjuntos dos caracteres restantes ignorando c (recursão)

Combinações - Algoritmo

subconjuntos(*pronto*, *restante*)

Se *restante* é vazio
 imprime(*pronto*)

Senão

$c \leftarrow \text{restante}[1]$ (*primeiro caracter da string*)

 subconjuntos(*pronto* + *c*, *restante* - *c*)

 subconjuntos(*pronto*, *restante* - *c*)

listar_subconjuntos(*string*)

subconjuntos("", *string*)

