

# Strategie řešení sníž a vyřeš

---

doc. Mgr. Jiří Dvorský, Ph.D.

Stav prezentace ke dni 28. dubna 2024

Katedra informatiky

Fakulta elektrotechniky a informatiky

VŠB – TU Ostrava



## Strategie řešení sníž a vyřeš

Třídění vkládáním – InsertSort

Topologické třídění

Generování kombinatorických objektů

Generování permutací

Generování podmnožin

Snížení o konstantní faktor

Snížení o proměnný faktor

Strategie řešení sniž a vyřeš

Třídění vkládáním – InsertSort

Strategie řešení sníž a vyřeš

Topologické třídění

# Strategie řešení sníž a vyřeš

## Generování kombinatorických objektů

# Generování kombinatorických objektů

- Generování kombinací, variací, permutací, podmnožin je součástí různých algoritmů.
- Typicky jde o výběr nějaké alternativy, možnosti, nastavení parametrů...
- Příklady – Problém obchodního cestujícího, Problém batohu.
- Matematika se zajímá spíše počty těchto objektů, zatímco informatika hledá algoritmy jak je generovat.
- **Počet těchto objektů roste exponenciálně nebo i rychleji!**

# Generování permutací

- Budeme generovat permutace celých čísel  $1, 2, \dots, n$ .
- Obecněji můžeme generovat permutací prvků  $\{a_1, a_2, \dots, a_n\}$ .
- Využití strategie sniž a vyřeš:
  1. Generování  $n!$  permutací pro  $n$  prvků převedeme na generování  $(n - 1)!$  permutací  $n - 1$  prvků.
  2. Jakmile máme problém pro  $n - 1$  vyřešen, vložíme prvek  $n$  na všech  $n$  možných pozic do každé z  $(n - 1)!$  permutací.
  3. Jinak řečeno, máme  $(n - 1)!$  permutací, ke každé z nich vygenerujeme  $n$  dalších. Celkově dostáváme  $n(n - 1)! = n!$  permutací.

## Generování permutací – příklad

---

permutace prvku 1	1		
vložení 2 do permutace 1 zprava doleva	12	21	
vložení 3 do permutace 12 zprava doleva	123	132	312
vložení 3 do permutace 21 zleva doprava	321	231	213

---

### Co je z příkladu patrné?

- Všechny permutace jsou navzájem různé.
- Minimální změna mezi permutacemi – dvě po sobě jdoucí permutace se liší výměnou jediné dvojice prvků a to dokonce sousedních prvků.



# Johnson-Trotterův algoritmus

- Existuje možnost generovat permutace  $n$  prvků? Bez nutnosti vygenerovat permutace pro  $n - 1$ ? Ano, existuje.
- Každému z  $n$  prvků permutace přiřadíme **šipku** (směr), buď vlevo nebo vpravo.
- O prvku  $k$  řekneme, že je v dané permutaci **mobilní**, jestliže sousední prvek ve směru šipky prvku  $k$  je menší než  $k$ .

## Příklad

Permutace se šipkami

3 2 4 1

Prvky 3 a 4 jsou mobilní, prvky 2 a 1 nejsou mobilní.

# Johnson-Trotterův algoritmus

**Vstup** : Přirozené číslo  $n$

**Výstup**: Seznam všech permutací čísel  $\{1, \dots, n\}$

```
1  $\pi \leftarrow \hat{1} \hat{2} \dots \hat{n}$ ;  
2 while  $\pi$  obsahuje mobilní prvek do  
3   |  $k \leftarrow$  největší mobilní prvek v  $\pi$ ;  
4   | přehod' v  $\pi$  prvek  $k$  se sousedem ve směru šipky;  
5   | změň směr šipky u všech prvků větších než  $k$ ;  
6   | vlož nově vytvořenou permutaci (krok 4)  $\pi$  do  
   |   výsledného seznamu;  
7 end  
8 return seznam všech permutací;
```

# Johnson-Trotterův algoritmus – příklad

Ukázka generování permutací pro  $n = 3$

1	2	3
1	3	2
3	1	2
3	2	1
2	3	1
2	1	3

O prvku  $k$  řekneme, že je v dané permutaci **mobilní**, jestliže sousední prvek ve směru šipky prvku  $k$  je menší než  $k$ .

# Johnson-Trotterův algoritmus – příklad, $n = 4$

1 2 3 4

1 2 4 3

1 4 2 3

4 1 2 3

4 1 3 2

1 4 3 2

1 3 4 2

1 3 2 4

3 1 2 4

3 1 4 2

3 4 1 2

4 3 1 2

4 3 2 1

3 4 2 1

3 2 4 1

3 2 1 4

2 3 1 4

2 3 4 1

2 4 3 1

4 2 3 1

4 2 1 3

2 4 1 3

2 1 4 3

2 1 3 4

# Johnson-Trotterův algoritmus

- Jeden z nejefektivnějších algoritmů pro generování permutací.
- Časová složitost algoritmu je  $\Theta(n!)$ .
- „Děsivá“ složitost algoritmu ale není způsobena algoritmem samotným, ten pracuje velice rychle. Je způsobena obrovským množstvím permutací, které musí vygenerovat...

Děkuji za pozornost