

# Domácí úkol k cvičení číslo 8

22. dubna 2024

## Generování permutací - Levitin str. 145

Permutace budeme generovat pomocí čísel se šipkami, např.  $\overset{\rightarrow}{5}, \overset{\leftarrow}{7}$ . Číslo se šipkou  $k$  je *mobilní*, pokud jeho šipka ukazuje na menší číslo vedle něj. Například, pro permutaci

$$\overset{\leftarrow}{3} \overset{\leftarrow}{2} \overset{\rightarrow}{4} \overset{\rightarrow}{1},$$

je prvek 4 mobilní, zatímco 3, 2 a 1 nejsou. S použitím pojmu mobilního prvku můžeme dát následující popis algoritmu Johnson-Trotter pro generování permutací.

---

**Algorithm 1** JohnsonTrotter( $n$ )

---

```
1: procedure JOHNSONTROTTER( $n$ )
2:   Inicializuj první permutaci jako  $\overset{\leftarrow}{1} \overset{\leftarrow}{2} \dots \overset{\leftarrow}{n}$ 
3:   while předchozí permutace obsahuje mobilní prvek do
4:     Najdi její největší mobilní prvek  $k$ 
5:     Proveďte výměnu prvku  $k$  s prvkem, na který ukazuje jeho šipka
6:     Otočte směry šipek všech prvků, které jsou větší než  $k$ 
7:     Přidejte novou permutaci do seznamu
8:   end while
9: end procedure
```

---

**Při implementaci proveďte rozumné rozdělení programu do funkcí.**

Použijte adekvátní datové struktury, ať vyrobíte čitelný program.

Zde je aplikace tohoto algoritmu pro  $n = 3$ :

$$\overset{\leftarrow}{1} \overset{\leftarrow}{2} \overset{\leftarrow}{3} \quad \overset{\leftarrow}{1} \overset{\leftarrow}{3} \overset{\leftarrow}{2} \quad \overset{\leftarrow}{3} \overset{\leftarrow}{1} \overset{\leftarrow}{2} \quad \overset{\rightarrow}{3} \overset{\leftarrow}{2} \overset{\leftarrow}{1} \quad \overset{\leftarrow}{2} \overset{\rightarrow}{3} \overset{\leftarrow}{1} \quad \overset{\leftarrow}{2} \overset{\leftarrow}{1} \overset{\rightarrow}{3}.$$

Srovnějte výstup Vašeho programu s tím, co dostanete pomocí `std::next_permutation`. Srovnějte rychlosti pro velká  $n$ .