

# AG1 - Cvičení IV

Tommy Chu

## 1 Zadání

Navrhněte algoritmus, který pro zadaných  $k$  seřazených posloupností s celkovým počtem prvků  $n$  vyrobí v čase  $\mathcal{O}(n \log k)$  jednu seřazenou  $n$ -prvkovou posloupnost (z původních prvků). Dokažte, že váš algoritmus je korektní a že skutečně pracuje v čase  $\mathcal{O}(n \log k)$ .

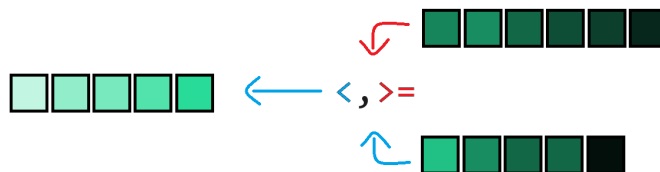
### 1.1 Pomocný algoritmus

#### Sloučení dvou seřazených posloupností

**Vstup:** dvě seřazené posloupnosti o celkem  $n$  prvcích

**Výstup:** jedna seřazená posloupnost z původních prvků

Porovnají se nejmenší prvky z obou posloupností a vybere se ten menší (rovnají-li se, tak libovolný), který se přemístí na konec finální posloupnosti, která je zpočátku prázdná. To se opakuje, dokud se obě posloupnosti nevyčerpají – pokud je jedna z posloupností prázdná, vybírá se automaticky z neprázdné. Finální posloupnost je seřazené sloučení vstupních posloupností. Algoritmus pracuje v čase  $\mathcal{O}(n)$ .



*Důkaz:* Zkonstruovaná posloupnost je nutně seřazená, protože v každém kroku se odebírá minimum ze zbývajících prvků. Protože jsou vstupní posloupnosti seřazené, lze nalézt nejmenší prvek v posloupnostech v čase  $\mathcal{O}(1)$  a algoritmus nepřekročí  $n$  porovnání, protože po každém porovnání se odebere alespoň jeden prvek. Algoritmus skutečně pracuje v čase  $\mathcal{O}(n)$ .

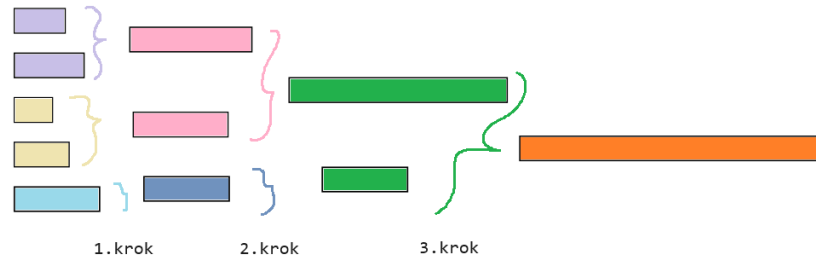
□

## 1.2 Řešení

**Vstup:**  $k$  seřazených posloupností o celkem  $n$  prvcích

**Výstup:** jedna seřazená posloupnost z původních prvků

V prvním kroku se posloupnosti rozdělí do  $\lceil \frac{k}{2} \rceil$  dvojic. Dvojice se sloučí popsáním algoritmem do jedné větší seřazené posloupnosti. V případě lichého počtu je jedna posloupnost sama ve dvojici a je automaticky ‘sloučená’. Krok se opakuje, dokud nezbyde jedna seřazená posloupnost.



V každém kroku dojde ke snížení počtu  $l$  nesloučených posloupností o polovinu na počet vytvořených dvojic  $\lceil \frac{l}{2} \rceil$  (počet potřebných sloučení se půlí). Počáteční počet posloupností je  $k$ , proto je potřeba  $\lceil \log k \rceil$  kroků. V každém kroku se slučují posloupnosti, které mají celkem  $n$  prvků, proto nám stačí maximálně  $n$  operací na sloučení všech dvojic.

Ke sloučení je tedy zapotřebí  $\lceil \log k \rceil$  kroků a v každém kroku nanejvýš  $n$  operací. Algoritmus tedy pracuje v čase  $\mathcal{O}(n \log k)$ .

□