Programovanie (1) v C/C++ 2025/26
Cvičenia 7, príklad 3
Merge

Algoritmus MergeSort z prednášky opakovane spája dve už utriedené postupnosti funkciou `merge`. Prvé volanie funkcie `merge` spája dve oblasti dĺžky jedna a postupne sa algoritmus dostáva k dlhším a dlhším oblastiam. Alternatívou je na začiatku nájsť vo vstupnom poli súvislé úseky, ktoré už sú utriedené a funkciu `merge` volať na takéto úseky. Vašou úlohou je **do priloženej kostry** doprogramovať takúto verziu MergeSortu.

Algoritmus na začiatku nájde vo vstupnom poli oblasti, ktoré sú už utriedené pomocou funkcie `najdiHotove`, ktorú máte naprogramovať. Tu je príklad vstupného poľa rozdeleného na 5 utriedených oblastí:

```
8 9 | 5 | 1 2 3 | 0 4 7 | 6 |
```

Funkcia `najdiHotove` má do poľa B uložiť konce týchto oblastí, t.j. indexy 1,2,5,8,9, kde napríklad index 1 je koniec oblasti $(8, 9)$ a index 8 je koniec oblasti $(0, 4, 7)$. Počet oblastí, t.j. v tomto príklade číslo 5, má uložiť do premennej m.

Keď algoritmus pole rozdelí na m utriedených oblastí, bude oblasti spájať funkciou `merge` z prednášky. Najskôr zlúči prvé dve oblasti, t.j. v našom prípade z $(8, 9)$ a (5) vznikne $(5, 8, 9)$. Potom zlúči druhé dve oblasti a podobne pokračuje až po koniec poľa. Ak bolo oblastí nepárny počet, posledná zostane v pôvodnom stave. V našom príklade teda ešte zlúči $(1, 2, 3)$ s $(0, 4, 7)$ do $(0, 1, 2, 3, 4, 7)$ a oblasť (5) nechá tak.

Celý proces hľadania a zlučovania oblastí sa niekoľkokrát opakuje, až kým počet oblastí m neklesne na jedna, čo znamená, že celé pole je jedna utriedená oblasť. Po nájdení oblastí sú tieto vždy vypísané vo formáte ako v príklade vyššie. Na rozdiel od MergeSortu z prednášky je tento program nerekurzívny. Vstup obsahuje počet triedených prvkov n a zoznam n celých čísel.

Priložená kostra už obsahuje načítanie, výpis, funkciu `merge` z prednášky, ako aj časť samotného algoritmu. Vašou úlohou je dopísať funkciu `najdiHotove` a chýbajúcu časť funkcie `tried`. **Nemeňte už hotové časti programu a dodržujte pokyny uvedené v komentároch.**

Príklad vstupu:

```
10
8 9 5 1 2 3 0 4 7 6
```

Príklad výstupu:

```
8 9 | 5 | 1 2 3 | 0 4 7 | 6 |
5 8 9 | 0 1 2 3 4 7 | 6 |
0 1 2 3 4 5 7 8 9 | 6 |
0 1 2 3 4 5 6 7 8 9 |
```

Príklad vstupu:

```
6
3 3 3 1 1 1
```

Príklad výstupu:

```
3 3 3 | 1 1 1 |
1 1 1 3 3 3 |
```