

Programovanie (1) v C/C++ 2023/24

Cvičenia 7, príklad 2

Quicksort

Algoritmus Quicksort z prednášky triedi oblasť poľa od pozície `left` po pozíciu `right`. Pokiaľ je dĺžka tejto oblasti väčšia ako 1, zvolí si prvý prvok oblasti ako pivot a následne aktuálnu oblasť poľa preusporiada tak, že v ľavej časti budú prvky menšie ako pivot, nasleduje pivot a za ním prvky väčšie alebo rovné pivotu. Potom sa spustí rekurzívne triedenie zvlášť na ľavú časť a na pravú časť, pričom pozícia pivota sa už do ďalších rekurzívnych triedení neposiela. Táto implementácia Quicksortu bude pomalá, ak sa niektoré hodnoty vo poli veľakrát opakujú. Ak si totiž ako pivota zvolíme takúto opakujúcu sa hodnotu, všetky ďalšie kópie pivota sa dostanú do pravej časti poľa a môže sa stať, že pravá časť tak bude oveľa väčšia oproti ľavej časti.

V tomto príklade máte do implementácie z prednášky spraviť malú zmenu, ktorá by v týchto prípadoch občas mohla pomôcť. Konkrétne po skončení funkcie `partition` zistíte, či hneď vedľa nájdenej pozície pivota nie je ešte niekoľko prvkov s tou istou hodnotou ako pivot, a ak áno, začnete pravú časť poľa až za nimi.

Napríklad, ak by po volaní funkcie `partition` boli v ľavej časti poľa hodnoty 2,1,1, nasledoval by pivot s hodnotou 4 a za ním prvky 4,4,5,4, tak dve štvorky idúce hneď za pivotom by sa z ďalších rekurzívnych volaní vynechali a do rekurzie by sa poslala ľavá časť 2,1,1 a pravá časť 5,4. Všimnite si, že kópie pivota, ktoré sa nachádzajú v pravej časti až za inými prvkami, sa nevynechávajú.

Priložená **kostra** načíta najskôr počet čísel n a zoznam čísel, ktoré bude triediť. Tieto čísla uloží do poľa, zavolá funkciu `quicksort` a na koniec vypíše výsledok. Vašou úlohou je spraviť dve malé zmeny do funkcie `quicksort`, **zvyšok programu nemeňte**. Prvá zmena je preskakovanie kópií pivota popísané vyššie.

V druhá zmena je pridanie niekoľkých výpisov vo formáte ako v príkladoch nižšie. Hneď po spustení funkcie `quicksort` sa vypíšu hodnoty `left`, `right` a zoznam prvkov v triedenej časti poľa (prvky môžete vypísať už hotovou funkciou `vypis`). V triviálnych prípadoch sa podá správa, že ide o triviálny prípad. A napokon pred ďalšími rekurzívnymi volaniami sa vypíše správa o tom, ktoré pozície budú vynechané z ďalšej rekurzie, lebo sa na nich nachádza pivot a ďalšie kópie tej istej hodnoty. Vypisujte prvú a poslednú vynechanú pozíciu. Váš program by sa mal správať presne rovnako ako v príkladoch vstupu a výstupu.

Príklad vstupu:

```
10
2 2 5 1 2 2 0 4 2 6
```

Príklad výstupu:

```
left: 0 right: 9 hodnoty: 2 2 5 1 2 2 0 4 2 6
pivoty 2 5
left: 0 right: 1 hodnoty: 0 1
pivoty 0 0
left: 0 right: -1 hodnoty:
trivialny pripad
left: 1 right: 1 hodnoty: 1
trivialny pripad
left: 6 right: 9 hodnoty: 5 4 2 6
pivoty 8 8
left: 6 right: 7 hodnoty: 2 4
pivoty 6 6
left: 6 right: 5 hodnoty:
trivialny pripad
left: 7 right: 7 hodnoty: 4
trivialny pripad
left: 9 right: 9 hodnoty: 6
trivialny pripad
vysledok:
0 1 2 2 2 2 2 4 5 6
```

Príklad vstupu:

6
3 3 3 1 1 1

Príklad výstupu:

left: 0 right: 5 hodnoty: 3 3 3 1 1 1
pivoty 3 5
left: 0 right: 2 hodnoty: 1 1 1
pivoty 0 2
left: 0 right: -1 hodnoty:
trivialny pripad
left: 3 right: 2 hodnoty:
trivialny pripad
left: 6 right: 5 hodnoty:
trivialny pripad
vysledok:
1 1 1 3 3 3