

Programovanie (1) v C/C++ 2022/23

Cvičenia 11, príklad 2

Vytvor strom

Priložená **kostra** pracuje s binárnym vyhľadávacím stromom, ktorý má vo vrcholoch ako dáta uložené celé čísla. Cieľom tejto úlohy je takýto strom vybudovať v prípade, že dostaneme všetky hodnoty naraz v poli utriedené od najmenšieho po najväčší. Mohli by sme síce prvky vkladať po jednom funkciou `insert`, ale namiesto toho chceme v tejto úlohe vytvoriť strom priamo algoritmom typu rozdeľuj a panuj, ktorý sa bude trochu podobat na algoritmy ako Merge Sort a rekurzívne binárne vyhľadávanie. Cieľom je vytvoriť strom tak, aby sa v každom vrchole čo najmenej líšila veľkosť (počet vrcholov) ľavého a pravého podstromu, lebo potom dostaneme strom s malou výškou, ktorý je výhodný pre rýchle vyhľadávanie.

Vašou úlohou je implementovať telo rekurzívnej funkcie s nasledujúcou hlavičkou:

```
node * buildSubtree(int left, int right, int a[], node *parent)
```

Táto funkcia dostane pole `a`, pričom z neho bude uvažovať iba úsek od indexu `left` po index `right`, vrátane. Funkcia pre tento úsek vybuduje strom a vráti smerník na jeho koreň. Funkcia dostane aj smerník `parent` na vrchol, ktorý má byť rodičom koreňa nového stromu, pričom túto hodnotu uloží do položky `parent` príslušného vrcholu. Funkcia má postupovať nasledovne:

- Ak je `right < left`, ide o úsek dĺžky nula (triviálny prípad). Výsledkom je prázdny strom.
- V opačnom prípade nájde index stredného prvku ako $(left + right) / 2$. Tento stredný prvok bude kľúčom v koreni, vytvorí teda príslušný vrchol.
- Napokon pomocou dvoch rekurzívnych volaní vytvorí ľavý podstrom pre zvyšné prvky naľavo od stredného prvku a pravý podstrom pre zvyšné prvky napravo od stredného prvku. V rekurzii nezabudnite správne nastaviť parameter `parent`.

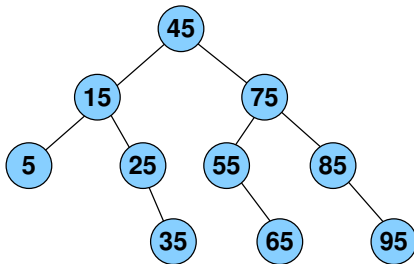
Kostra načíta počet prvkov n ($0 \leq n \leq 1000$) a rastúcu postupnosť n celých čísel, ktoré uloží do poľa. Potom vytvorí strom vašim algoritmom a nakoniec tento strom vypíše v inorder poradí, pričom pre každý uzol stromu vypisuje dáta v tomto uzle, ale pre kontrolu aj dáta v jeho rodičovi a deťoch, pričom ak rodič alebo niektoré dieťa neexistuje, vypíše namiesto hodnoty slovo `NULL`.

Váš kód píšete iba do funkcie `buildSubtree`. Nemeňte už hotové časti programu, nepridávajte ďalšie funkcie.

Príklad vstupu

```
10
5 15 25 35 45 55 65 75 85 95
```

Výsledný strom



Príklad výstupu

```
data: 5, parent: 15, left: NULL, right: NULL
data: 15, parent: 45, left: 5, right: 25
data: 25, parent: 15, left: NULL, right: 35
data: 35, parent: 25, left: NULL, right: NULL
data: 45, parent: NULL, left: 15, right: 75
data: 55, parent: 75, left: NULL, right: 65
data: 65, parent: 55, left: NULL, right: NULL
data: 75, parent: 45, left: 55, right: 85
data: 85, parent: 75, left: NULL, right: 95
data: 95, parent: 85, left: NULL, right: NULL
```