Programovanie (1) v C/C++ 2024/25
Cvičenia 12, príklad 4
Predchodca

Priložená **kostra** pracuje s binárnym vyhľadávacím stromom, ktorý má vo vrcholoch ako kľúče uložené celé čísla, pričom žiaden kľúč sa v strome neopakuje. Navyše má v každom vrchole aj smerníky `prev` a `next`, pričom `prev` ukazuje na uzol s najbližšou menšou hodnotou ako aktuálny uzol (predchodca) a `next` ukazuje na uzol s najbližšou väčšou hodnotou (následník). Ak by sme teda začali v uzle s najmenšou hodnotou a pohybovali by sme sa po smerníkoch `next`, prešli by sme všetky uzly v poradí podľa rastúcej hodnoty kľúča. Podobne ak začneme vo vrchole s maximálnym kľúčom, tak pomocou smerníkov `prev` prejdeme uzly v poradí podľa klesajúcej hodnoty kľúča. Smerník `prev` je `NULL` pre uzol s minimálnym kľúčom a smerník `next` pre uzol s maximálnym kľúčom v strome.

Tieto smerníky využijeme v operáciách `printPrev` a `printNext`, ktoré dostanú hodnotu `x` a `k` a vypíšu `k` najbližších následníkov alebo predchodcov vrchola s kľúčom `x`. Ak má kľúč `x` menej ako `k` následníkov respektíve predchodcov, vypíšu ich menší počet. Ak sa kľúč v strome nenachádza, nevypíšu nič.

Tieto nové funkcie sú už v kostre hotové, **vašou úlohou** je doprogramovať správne nastavenie smerníkov `next` a `prev` pri vkladaní nového uzla do stromu. Pozor, smerníky `prev` a `next` treba nastaviť nielen novo-vloženému uzlu, ale aj jeho predchodcovi a následníkovi. Predchodcu a následníka nemusíte v strome vyhľadávať algoritmom z prednášky, nakoľko pri vkladaní nového uzlu je jeho rodič vždy buď predchodcom alebo následníkom, v závislosti od toho, či je vkladaný uzol pravým alebo ľavým dieťaťom. Ak poznáme napríklad predchodcu nového uzla `v`, tak následník uzla `v` bude ten uzol, ktorý bol doteraz následníkom tohto predchodcu.

Kostra načítava a vykonáva príkazy `add`, ktorý do stromu pridá nový kľúč, `prev`, ktorý vypíše požadovaný počet predchodcov, `next`, ktorý vypíše požadovaný počet následníkov a `end`, ktorý ukončuje vstup.

Váš kód píšte na miesta vyznačené komentárom TODO. Nemeňte už hotové časti programu. Ak treba, pridajte ďalšie funkcie.

Pomôcka: Na ďalšej strane je obrázok so všetkými smerníkmi v malom strome a ako sa zmenia pridaním uzla. Bolo by však ideálne, ak by ste si takýto obrázok vedeli na základe zadania nakresliť aj sami, takže obrázok si pozrite len v prípade potreby. Podobné obrázky vám môžu pomôcť pri riešení príkladov so smerníkmi.

Príklad vstupu

```
prev 50 2
next 50 2
add 50
add 30
add 40
prev 30 3
next 30 3
prev 40 3
next 40 3
prev 50 3
next 50 3
add 60
add 80
next 30 10
prev 80 10
end
```

Príklad výstupu

```
up to 2 predecessors of 50:
up to 2 successors of 50:
adding 50
adding 30
adding 40
up to 3 predecessors of 30:
up to 3 successors of 30: 40 50
up to 3 predecessors of 40: 30
up to 3 successors of 40: 50
up to 3 predecessors of 50: 40 30
up to 3 successors of 50:
adding 60
adding 80
up to 10 successors of 30: 40 50 60 80
up to 10 predecessors of 80: 60 50 40 30
```

Naľavo je strom s kľúčmi 30 a 50, napravo ten istý strom po vložení kľúča 40. Červenou farbou sú zvýraznené smerníky, ktoré sa zmenili vložením kľúča. Z nich smerník `right` v uzle 30 je nastavený v kostre, ostatné štyri potrebujete nastaviť vy. Položky bez šípok sú smerníky s hodnotou `NULL`. Smerník `parent` pre jednoduchosť nezobrazujeme.