Podatkovne strukture in algoritmi
(2024/25)

2. domača naloga – teoretični del

1. naloga: Patricijina drevesa. V tej nalogi predpostavimo, da smo vstavili $n \geq 2$ elementov v Patricijino drevo nad abecedo Σ in da noben element ni predpona kakšnega drugega.

(i) Če je $\Sigma = \{0, 1\}$, pokažite, da ima Patricijino drevo natanko $n - 1$ notranjih vozlišč.

(ii) Če je $\Sigma = \{0, 1, X\}$, kakšna je največja možna višina v Patricijinemu drevesu in kakšna najmanjša? Utemeljite.

2. naloga: Dvojiška drevesa. V tej nalogi višino drevesa definiramo kot število vozlišč, vključno s korenom, na najdaljši poti od korena do lista.

(i) Koliko največ vozlišč ima lahko dvojiško drevo višine h? Utemeljite.

(ii) Koliko najmanj vozlišč ima lahko dvojiško drevo višine h? Utemeljite.

(iii) Koliko je največja možna višina dvojiškega drevesa z n vozlišči? Utemeljite.

(iv) Koliko je najmanjša možna višina dvojiškega drevesa z n vozlišči? Utemeljite.

3. naloga: AVL drevesa.

(i) Narišite AVL drevo, ki ima najmanjšo možno višino in n vozlišč, za $n = 3, 5, 7$.

(ii) Pokažite, da ima AVL drevo z n elementi višino $O(\log n)$.

4. naloga: B-drevesa. V tej nalogi višino drevesa definiramo kot število vozlišč, vključno s korenom, na najdaljši poti od korena do lista.

(i) Koliko je najmanj elementov v B-drevesu reda 5 in višine $h = 1, 2, 3, 5$? Utemeljite.

(ii) Koliko je najmanj elementov v B-drevesu reda b in višine h? Poščite zaprto formulo.

5. naloga: B-drevesa.

(i) Recimo, da iskanje v B-drevesu znotraj vsakega vozlišča implementiramo z uporabo bisekcije namesto z linearnim pregledom. Pokažite, da v tem primeru iskanje elementa zahteva $O(\log n)$ primerjav, kjer je n število elementov v strukturi.

(ii) Naj bosta T in U B-drevesi reda 4 in naj velja, da so vsi elementi v T manjši od elementa x, vsi elementi v U pa večji od x. Recimo, da T premore n elementov, U pa m elementov. Opišite algoritem s časovno zahtevnostjo $O(\log n + \log m)$, ki združi T in U v eno B-drevo $T \cup U$ reda 4 z $m + n + 1$ elementi ($T \cup U$ sestoji iz elementa x in vseh elementov v T in U).

6. naloga: Rdeče črna drevesa. Naj bo $\mathcal{D}$ podatkovna struktura na množici pozitivnih celih števili, ki podpira naslednje operacije:

- **Vstavi**$(\mathcal{D}, x)$: vstavi x v $\mathcal{D}$.
- **Odstrani**$(\mathcal{D}, x)$: odstrani x iz $\mathcal{D}$.
- **Poišči**$(\mathcal{D}, x)$: najde x v $\mathcal{D}$.
- **Preštej**$(\mathcal{D}, x)$: vrne število elementov v $\mathcal{D}$, ki so večji od x.

Podatkovno strukturo $\mathcal{D}$ bi radi implementirali z uporabo rdeče črnih dreves. Opišite, kako lahko prilagodimo rdeče črno drevo, da bodo časovne zahtevnosti vseh štirih operacij $O(\log n)$, kjer je n število elementov v strukturi. Utemeljite, zakaj je časovna zahtevnost vseh štirih operacij $O(\log n)$.

Rešitev nalog oddajte preko e-učilnice: **oddajte .pdf datoteko**. Vse naloge je potrebno reševati **samostojno**. Prepisovanje se kaznuje z negativnimi točkami. Rok za oddajo nalog je **nedelja, 24. november** 2024.