

Podatkovne strukture in algoritmi (2024/25)

3. domača naloga – programerski del

Razpršene tabele

Na strežniku za oddajanje nalog (<http://marmoset.famnit.upr.si>) je naložena tretja domača naloga, pri kateri je potrebno sprogramirati razpršeni tabeli. Sledite naslednjim navodilom:

- V razredu `HashTable` implementirajte razpršeno tabelo in upoštevajte:
 - (i) Ključi so cela števila (torej lahko tudi negativna!) in jih shranjujete v tabelo `int[] data`.
 - (ii) Za funkcijo zgoščanja uporabite funkcijo $h(x) = (x \cdot 701) \bmod 2000$. Velikost tabele je torej 2000. Bodite pozorni, ko računate indeks negativnega ključa. Na primer, kjuču $x = -10$ pripada indeks

$$h(-10) = (-7010) \bmod 2000 = (2000(-4) + 990) \bmod 2000 = 990.$$

Indeks mora torej biti vedno nenegativno celo število med vključno 0 in 1999!

- (iii) Implementirajte metodo `insert`, ki sprejme celo število in ga vstavi v tabelo. Metoda vrne `true`, če je bilo vstavljanje uspešno. Če je število že v tabeli, mora funkcija `insert` vrniti `false`.
 - (iv) Implementirajte metodo `search`, ki sprejme celo število in ga poišče v tabeli. Metoda vrne `true`, če je bilo iskanje uspešno in `false` sicer.
 - (v) Implementirajte metodo `delete`, ki sprejme celo število in ga izbriše iz tabele. Metoda vrne `true`, če je bilo brisanje uspešno in `false` sicer.
 - (vi) Sovpadanje rešujete z veriženjem. Toda ne kot običajno s povezanim seznamom, ampak z razpršeno tabelo, ki jo implementirate v razredu `HashTable2` (kombinirana struktura).
- V razredu `HashTable2` implementirajte razpršeno tabelo in upoštevajte:
 - (i) Ključi so cela števila (torej lahko tudi negativna!) in jih shranjujete v tabelo `int[] data`.
 - (ii) Za funkcijo zgoščanja uporabite funkcijo $h(x) = (x \cdot 53) \bmod 100$. Velikost tabele je torej 100. Kot zgoraj bodi pozorni, da pravilno izračunate indeks, ko delate z negativnim ključem. V tem primeru mora biti indeks vedno nenegativno celo število med vključno 0 in 99!

- (iii) Implementirajte metodo `insert`, ki sprejme celo število in ga vstavi v tabelo. Metoda vrne `true`, če je bilo vstavljanje uspešno. Če je število že v tabeli, mora funkcija `insert` vrniti `false`. Če vstavljate element v polno tabelo, mora metoda `insert` vrniti `false`.
- (iv) Implementirajte metodo `search`, ki sprejme celo število in ga poišče v tabeli. Metoda vrne `true`, če je bilo iskanje uspešno in `false` sicer.
- (v) Implementirajte metodo `delete`, ki sprejme celo število in ga izbriše iz tabele. Metoda vrne `true`, če je bilo brisanje uspešno in `false` sicer.
- (vi) Sovpadanje rešujete z linearnim naslavljanjem. Bodite pozorni, ko brišete element. Potrebujete poseben znak, ki označuje, da ste element zbrisali, zato, da bo funkcija `search` pravilno delovala. Prav tako bodi pozorni, ko vstavljate nov element, saj ga lahko vstavite tudi na mesto, ki je označeno s posebnim znakom (ki označuje mesto zbrisanega elementa).

Rešitev naloge oddajte preko <http://marmoset.famnit.upr.si>. Vsi testi, tako javni kot skriti, so točkovno enakovredni. Rok za oddajo nalog je **nedelja, 8. december 2024**.