

Algoritmi in podatkovne strukture – 2

Osnovne podatkovne strukture

Osnovne podatkovne strukture

- implicitne in eksplicitne podatkovne strukture
- polje
- vrsta, sklad
- povezan seznam
- hierarhične in linearne podatkovne strukture

Podatkovne strukture

- podatki imajo obliko: (key, data)
- podatkovna struktura je način organiziranja podatkov, da do njih lažje dostopamo
- pomembne so (vsaj) naslednje lastnosti strukture:
 - pravilnost
 - časovna zahtevnost operacij nad strukturo
 - prostorska zahtevnost operacij nad strukturo
- eden od načinov razlikovanja med podatkovnimi strukturami je:
 - implicitne podatkovne strukture:** dostop do podatkov strukturi se izvaja samo na podlagi vrednosti ključa; primer je *polje*
 - eksplicitne podatkovne strukture:** dostop do podatkov se izvaja na podlagi primerjanj vrednosti ključev; primer je *drevo*

Primer

- Imamo univerzalno množico ključev

$$\mathcal{M} = \{0, 1, \dots, m - 1\}$$

in v njej podmnožico

$$\mathcal{N} = \{a_0, a_1, \dots, a_{n-1}\} \subseteq \mathcal{M}$$

Imamo še operacijo $\text{Find}(\mathcal{N}, a)$, ki ugotovi ali je $a \in \mathcal{N}$.

- rešitev 1: naštejemo vse elemente iz \mathcal{N} in po vrsti primerjamo
- rešitev 2: imamo polje `odg` Boolovih spremenljivk dolžine m in vrednost `odg[a]` pove ali je element v \mathcal{N}
- zahtevnosti:

rešitev	čas	prostor
1.	$O(n)$	$O(n)$
2.	$O(1)$	$O(m)$

- Primer: $\mathcal{M} = \{0, 1, \dots, 15\}$ in $\mathcal{N} = \{2, 6, 7, 13\}$

Osnovni strukturi

- imamo operaciji: Insert in Delete
- vrsta (FIFO): pri brisanju izbrišemo element, ki je najdlje časa že v strukturi
- sklad (LIFO): pri brisanju izbrišemo element, ki je najmanj časa v strukturi

Osnovne implicitne podatkovne strukture

- polje: dostop preko indeksa
- sklad: dodatna informacija, zadnji poln element
- vrsta: dodatne informacije
 - prvi in zadnji poln element
 - prvi in zadnji poln element ter ali je vrsta polna ali prazna
- zahtevnosti:

rešitev	čas	prostor
sklad	$O(1)$	$n + 1$
vrsta 1	$O(1)$	$(n - 1) + 2$
vrsta 2	$O(1)$	$n + 2 + 1\text{bit}$

Osnovne eksplicitne podatkovne strukture

- povezan seznam: izrecno zabeleženo mesto (naslov, referenca) nahajanja podatka
- sklad: dodatna informacija, prvi element
- vrsta: dodatni informaciji, prvi in zadnji element
- zahtevnosti:

rešitev	čas	prostor
sklad	$O(1)$	$2n + 1$
vrsta	$O(1)$	$2n + 2$

Statične in dinamične podatkovne strukture

- imamo množici \mathcal{M} in \mathcal{N}
- običajno se \mathcal{M} ne spreminja, \mathcal{N} pa se spreminja
 - dinamična struktura: dodajanje in odzemanje
 - poldinamična struktura: se samo dodaja
- kaj pa iskanje najbolj podobnega elementa?
 - statične: lahko vnaprej pripravimo podatke – implicitna podatkovna struktura izgleda idealno
 - (pol)dinamične: sproti popravljamo podatke – implicitna podatkovna struktura ne izgleda sploh primerna

Hierarhične podatkovne strukture

- zasnovane na pojmu *rekurzije*
- operacija:
 - naredi *tukaj* in, če se ne dâ
 - naredi na preostanku strukture (ali delu strukture)
- primer: povezan seznam, drevo, številsko drevo, ...