Univerza v Ljubljani

Fakulteta za računalništvo in informatiko

Igor Rožanc

Osnove algoritmov in podatkovnih struktur I

(OAPS I)

2. letnik, VSP Računalništvo in informatika, vse smeri

PREDVIDENE PROSOJNICE ZA NASLEDNJA PREDAVANJA

Študijsko leto 2007/08

Sortiranje tabel – Quicksort

Iterativna rešitev:

- izdelamo seznam zahtev po porazdelitvah, ki še niso bile opravljene
- vsakič nastaneta 2 zahtevi:
  - eno (levo) obdelamo takoj,
  - drugo (desno) umestimo na seznam
- zahtevke obravnavamo v obratnem vrstnem redu: **utripajoč sklad**

Ponazoritev zahtevka: leva in desna meja dela tabele

Ponazoritev sklada: tabela objektov tipa `ElementSklada`

Razred ElementSklada ...

Prikaz delovanja algoritma ...

Sortiranje tabel – Quicksort 32

Ralizacija metode v Javi: metoda QuicksortI ...

Problem velikosti sklada

Ralizacija metode v Javi: popravljena metoda QuicksortI ...

Primer:

- dopolnitev razreda Sortiranje objektov (z metodo QuicksortI),
- sprememba razreda GlavniProgram ...

Primer alternativne rešitve:

- dopolnitev razreda Sortiranje objektov (z metodo QuicksortI1),
- sprememba razreda GlavniProgram ...

Dodatna optimizacija metode:

- izbira "pravega" srednjega elementa
- kombinacija z navadno metodo

Sortiranje tabel – Bucketsort 33

Posplošitev ideje deli in vladaj do skrajnosti: problem delimo na toliko delov/košev, kolikor je elementov ...

Primeren, ko imamo veliko zalogo vrednosti (recimo n-mestna števila)

1. Pripravi (pomožne) tabele (zbirke) za n košev
2. Določi najmanjši in največji element (**`min`** in **`max`**)
3. Porazdeli elemente po koših:

   `a[i]` sodi v koš j, kjer je **`j= n*(a[i]-min)/(max-min+1)`**
4. Sortiraj vsak koš posebej (s kakršnokoli metodo)
5. Sestavi koše (po zaporedju) zopet v tabelo **`a`**

Prikaz delovanja algoritma ...

Težave: - iskanje najmanjšega in največjega elementa

- izvedba n zbirk s tabelami

Sortiranje tabel – Bucketsort 34

Ralizacija metode v Javi: metoda Bucketsort ...

Primer:

- dopolnitev razreda Sortiranje objektov (z metodo Bucketsort),
- dopolnitev razreda Student (z metodo Vrednost)
- sprememba razreda GlavniProgram ...

Analiza časovne kompleksnosti:

- število košev n: O(n)
- koši imajo v povprečju enega (ali le nekaj elementov)
- **Skupaj: T = O(n),** vendar slabše kot Quicksort
- potrebujemo veliko dodatnega prostora !!!

Sortiranje tabel – BucketsortK 35

Izboljšava:

1. Pripravi (pomožne) tabele (zbirke) za fiksno število **K** košev

...

4. Sortiraj vsak koš posebej (**rekurzivno** z metodo **`BucketsortK`**)

...

Ralizacija metode v Javi: metoda BucketsortK ...

Primer:

- dopolnitev razreda Sortiranje objektov (z metodo BucketsortK),
- sprememba razreda GlavniProgram ...

Analiza časovne kompleksnosti:

- število košev K
- število rekurzivnih klicev: $O(\log_K n)$
- K=2: Quicksort
- **Skupaj: $T = O(n*\log_K n)$**

Leksikografsko sortiranje 36

Omejitev:

- deli atributa (ključa) za urejanje imajo pomen
- uporabno za sortiranje **števil** in **nizov** (znakov)

Postopek:

1. elemente tabele razvrstimo v več razredov glede na vrednost dela atributa (mesto v ključu)
2. postopek ponavljamo za vse dele (mesta) od najmanj do najbolj pomembnega
3. vsaka iteracija elemente najprej porazdeli in nato spet prepiše v tabelo `a`

Prikaz delovanja algoritma za sortiranje trimestnih števil ...

Ralizacija v Javi: razreda LeksiSort in TestLeksiSort ...

Leksikografsko sortiranje 37

Primer za sortiranje števil:

- razreda LeksiSort in TestLeksi Sort...

Spremembe (dodatki) za sortiranje nizov...

Prikaz delovanja algoritma za sortiranje nizov ...

Ralizacija v Javi: razreda LeksiSort1 in TestLeksiSort1 ...

Primer za sortiranje nizov:

- dopolnitev razreda LeksiSort in TestLeksi Sort...

Analiza časovne kompleksnosti:

- upoštevamo število premikov, primerjave niso primerljive
- navidez zelo dobro: O(n)
- dejansko le redko boljše od quicksorta, ker ne upoštevamo dodatnega dela

Vmesnik (ang. interface) 38

Nekateri programski jeziki omogočajo **večkratno dedovanje:** podrazred podeduje atribute in metode več kot enega nadrazreda

Problemi:

- atributi in metode v nadrazredih imajo enaka imena
- `super()` – klic konstruktorja iz katerega nadrazreda?

Java ne omogoča večkratnega dedovanja

Namesto tega ponuja koncept vmesnika

Razlika med vmesnikom in razredom:

- vse metode v vmesniku morajo biti `abstract`
- vsi atributi (če jih ima) morajo biti `static final`

Vmesnik predpiše metode, ki jih mora implementirati nek podrazred

=> predpiše obnašanje podrazreda

Vmesnik (ang. interface) 39

Podrazred lahko deduje samo od enega nadrazreda, implementira pa lahko več različnih vmesnikov.

```
public class Podrazred extends Nadrazred implements
                                        Vmesnik1, Vmesnik2
```

S stališča dedovanja `Podrazred`:

- podeduje atribute in metode razreda `Nadrazred`
- dodatno deklarira nove atribute in metode
- če mu katera od podedovanih metod ne ustreza, jo lahko redefinira

S stališča implementacije vmesnika `Podrazred`:

- deklarira vse metode, ki so specificirane v vmesnikih `Vmesnik1`, `Vmesnik2`
- lahko uporblja atribute (statične spremenljivke), ki so definirani v vmesnikih

Podrazred je razširitev treh tipov: Nadrazred, Vmesnik1 in Vmesnik2

Primerjava: abstraktni razred - vmesnik

Enako:

- ne moremo generirati objektov abstraktnega tipa ali vmesnika

Različno:

- v abs.razredih so lahko samo nekatere metode abstraktne, v vmesniku so vse
- atributi abs.razreda se obnašajo kot spremenljivke objektov, atributi vmesnika pa so statične konstante.

Uporaba abstraktnega razreda:

- vnaprej deklariramo znane atribute in metode, ki jih podeduje podrazred: recimo pri igri s kartami metodo `mesaj()`

Uporaba vmesnika:

- vmesnik določa le del značilnosti, ki jih vsak podrazred implementira na svoj način: recimo pri glasbenih instrumentih metodo `zaigrajTon()`