

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

PROSOJNICE ZA 7. PREDAVANJA (15.11.2007)

Študijsko leto 2007/08

Sortiranje tabel – Quicksort

31

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 `ElementSkлада`

Razred ElementSkлада ...

Prikaz delovanja algoritma ...

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

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:

$$a[i] \text{ sodi v koš } j, \text{ kjer je } j = n * (a[i] - \text{min}) / (\text{max} - \text{min} + 1)$$

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

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 !!!**