

Algoritmi in podatkovne strukture (Teorija)

1. naloga Izpeljite asimptotično (Θ) zahtevnost funkcije $f(n) = \sum_{k=1}^n k$.

2. naloga Problem P rešujemo z metodo *deli in vladaj* takole: P razdelimo v podprobleme (iste vrste kot P), od katerih je vsak velikosti $n/2$. Sedem podproblemov rešimo in iz njihovih rešitev sestavimo rešitev problema P . Priprava podproblemov in sestavljanje njihovih rešitev v končno skupaj terjata $5n^2$ časa. Kakšna je asimptotična časovna zahtevnost tega algoritma? Odgovor utemelji.

3. naloga Za vsakega od spodnjih algoritmov povejte: a) Kakšen problem rešuje? b) Kakšno časovno zahtevnost $O(f(n))$ ima? Pri vsakem odgovoru povejte, kaj pomeni n .

1. Dvojiško iskanje

a)

b)

2. Quicksort

a)

b)

3. Heapsort

a)

b)

4. Bubblesort

a)

b)

5. Strassenovo množenje

a)

b)

4. naloga Zapišite formalno definicijo relacije $f(n) = \mathcal{O}(g(n))$, kjer sta f in g funkciji.

5. naloga Izpeljite asimptotično (Θ) zahtevnost funkcije $f(n) = e^n n!$ Katera od funkcij $f(n)$ in $g(n) = n^{n+1}$ asimptotično hitreje narašča?

6. naloga (a) Kakšna je asimptotična časovna zahtevnost (Θ) hitrega algoritma za iskanje k -tega elementa po velikosti med n urejenimi števili?
(b) Opiši, kako pri tem algoritmu uporabimo metodo deli in vladaj.

7. naloga (b) Kakšna bi bila asimptotična časovna zahtevnost Quicksorta, ki bi za delilne elemente jemal mediane? (Mediana je po velikosti srednji element.) Odgovor utemelji.

8. naloga V čem je razlika med metodo deli in vladaj in dinamičnim programiranjem?

9. naloga Izpeljite asimptotično (Θ) zahtevnost funkcije $f(n) = a^{\log_b n}$, kjer sta $a, b > 1$ konstanti.

10. naloga Problem P bomo reševali z algoritmom po metodi *deli in vladaj* takole: P razdelimo v podprobleme (iste vrste kot P), od katerih je vsak velikosti $n/2$. Sedem od podproblemov rešimo in iz njihovih rešitev sestavimo rešitev problema P . Priprava podproblemov in sestavljanje njihovih rešitev skupaj terjata $5n^2$ časa. Kakšna je asimptotična časovna zahtevnost tega algoritma? Odgovor utemelji.

Algoritmi in podatkovne strukture (Praksa)

1. naloga

Podano imate sled izvajanja različice nekega (znanega) algoritma za urejanje na tabeli števil 33, 2, 11, 9, 17, 4, 15, 22.

```
33 2 11 9 17 4 15 22 p=14
2 11 9 4 | 33 17 15 22
2 11 9 4 p=6
2 4 | 11 9
2 4 p=3
2 | 4
33 17 15 22 p=21
17 15 | 33 22
17 15 p=16
15 | 17
33 22 p=27
22 | 33
```

1. Sled katerega algoritma je to? Napišite natančno obrazložitev in pravila, ki se uporabljajo pri izvajanju predstavljene različice algoritma.
2. Na zgornjem primeru algoritem razdeli tabelo na dva popolnoma uravnotežena dela. Zapišite primerek tabele dolžine 6 kjer se ob izvajanju enakega postopka razbijanje tabele izrodi, t.j., ena podtabela ima po delitvi dolžino 1.
3. Zapišite sled takega izrojenega primera.
4. Zapišite splošno tabelo (s parametrom n), ki bo predstavljala najslabši možni primer za zgoraj zapisani algoritem.

2. naloga Podan je spodnji algoritem.

```
function izpit(list, l, r)
  if r - l <= 1 then
    return max(list[r], list[l])
  else
    m = (r - l + 1) / 4
    a = izpit(list, l, l + 2 * m - 1)
    b = izpit(list, l + m, l + 3 * m - 1)
    c = izpit(list, l + 2 * m, r)
    return a + b + c
end
```

- a) Zapišite drevo izvajanja zgornjega algoritma na seznamu števil 2, 4, 5, 9, 7, 5, 4, 1.
- b) Z uporabo Master Theorem izračunajte njegovo časovno zahtevnost.