črt),nato pa naj si izpisi sledi sledijo: pri navadnih metodah izpišemo celotno zaporedje na koncu vsake izvedbe vsake zunanje zanke, črta pa ločuje urejeni del zaporedja od neurejenega dela zaporedja,

• pri urejanju s kopico najprej izpišemo zaporedje, ko zgradimo kopico, nato pa vsakič, ko kopico popravimo; črta deli kopico v začetnem delu zaporedja in zadnji urejeni del zaporedja. • pri urejanju z zlivanjem izpišemo del zaporedja, ki se trenutno ureja, črta pa deli levo in desno polovico pri zlivanju. Izpis se izvede ob delitvi na dva dela ter po zlivanju obeh delov.

• pri hitrem urejanju izpišemo le del zaporedja, ki ga trenutno urejamo in sicer vedno neposredno po porazdelitvi. Črte označujejo mesta delitve zaporedja pri naslednjem rekurzivnem klicu. Na koncu se izpiše še urejeno zaporedje (brez črt). pri korenskem urejanju izpišemo zaporedje po urejanju po vsakem mestu (enicah, deseticah, ...) brez vmesnih črt.

• pri urejanju s koši izpišemo zaporedje po razporeditvi v n/2 košev (med posameznimi koši naj bo črta) ter nato pri urejanju z navadnim vstavljanjem (vseh košev naenkrat) po vsaki izvedbi zunanje zanke (črta ločuje urejeni in neurejeni del).

Primeri (skupaj s klici):

trace insert up

42 17 27 51 39

17 42 | 27 51 39 17 27 42 | 51 39 17 27 42 51 | 39

42 17 27 51 39

17 27 39 42 51 trace select down 42 17 27 51 39

51 | 17 27 42 39 51 42 | 27 17 39 51 42 39 | 17 27

42 17 27 51 39

51 42 39 27 | 17 trace bubble down 42 17 27 51 39

> 42 17 27 51 39 51 | 42 17 27 39

51 42 39 | 17 27 51 42 39 27 | 17

trace heap up 42 17 27 51 39

> 42 17 27 51 39 51 42 27 17 39

42 39 27 17 | 51 39 17 27 | 42 51 27 17 | 39 42 51 17 | 27 39 42 51

42 17 27 51 39 42 17 27 51 39

trace merge up

42 17 27 | 51 39 42 17 | 27 42 | 17 17 42 17 27 42 51 | 39

17 27 39 42 51

42 17 27 51 39 51 | 42 | 27 17 39

trace quick down 42 17 27 51 39

39 51

39 | 27 | 17 51 42 39 27 17 trace radix up

42 17 27 51 39 42 17 27 51 39

51 42 17 27 39 17 27 39 42 51

42 17 27 51 39 42 17 27 51 39

trace bucket down

42 51 39 | 17 27 51 42 | 39 17 27

51 42 39 | 17 27 51 42 39 17 | 27 51 42 39 27 17

Poleg sledenja mora program omogočati tudi štetje števila premikov (tj. prirejanj elementov, recimo metoda swap() izvede 3 premike) in števila primerjav elementov (v primeru zadnjih dveh metod štejemo k primerjavam tudi število določanj mesta elementa v tabeli števcev c recimo število klicev metode getDigit(), če je to izvedeno s to metodo). V ta namen je treba v program vgraditi ustrezne števce. Sprožitev tega načina delovanja je enaka kot v primeru sledenja, le ukaz je count(). Pri tem se izpišejo vrednosti števila premikov in primerjav za tri zaporedne izvedbe urejanja z izbrano metodo in sicer:

Štetje premikov in primerjav

 za urejanje podanega zaporedja v izbrani smeri, za urejanje že urejenega zaporedja v izbrani smeri in

za urejanje že urejenega zaporedja v obratni smeri.

Vrednosti so zapisane ena za drugo ločene s presledkom, med tremi urejanji je črta ().

Primeri: count insert up

42 17 27 51 39

count select down 42 17 27 51 39

12 7 | 8 4 | 18 10

12 10 | 12 10 | 12 10

count bubble down 42 17 27 51 39

18 8 | 0 4 | 30 10

count heap up 42 17 27 51 39

42 17 27 51 39

24 12 | 30 12 | 24 10

14 10 | 16 18 | 16 16

20 20 | 20 20 | 20 20

20 22 | 18 18 | 22 21

24 8 | 24 7 | 24 5

count quick down

42 17 27 51 39

count radix up 42 17 27 51 39

count bucket down 42 17 27 51 39

Oddaja Vse razrede in vmesnike združite v eno datoteko z imenom Naloga2.java, ki jo shranite/oddate v pod opcijo Oddaj nalogo 2 - Urejanje zaporedja števil.

Veliko uspeha pri delu!

Rok za oddajo: do nedelje, 30. decembra 2018 (do 23.59).

→ Oddaj nalogo 1 - Programabilni kalkulator

Status oddaje naloge

Pri tej nalogi vam ni treba oddati ničesar. Status oddaje naloge Stanje ocen Neocenjeno Rok za oddajo nedelja, 30. december 2018, 23:59 Preostali čas Rok za oddajo naloge je potekel Zadnja sprememba

\$

Naloga 3: Algoritmi nad grafi -

Komentar oddaje ★ Komentarji (0)

Skok na...