

Ariketa01

Idatzi erabiltzaileari bi zenbaki oso eskatu eta bi zenbaki horien Zatitzaile Komun Handiena kalkulatzeko duen programa. Erabili eta moldatu azpiprogramen ataleko ariketetako erabilitako kodea (3. unitateko [Aziprogramak eta datu-egitura estatikoak] 1. ariketa multzoko 2. ariketa).

Ariketa02

Idatzi erabiltzaileari bi zenbaki oso eskatu eta bi zenbaki horien Multiplo Komun Txikiena kalkulatzeko duen programa. Erabili eta moldatu azpiprogramen ataleko ariketetako erabilitako kodea (3. unitateko [Aziprogramak eta datu-egitura estatikoak] 1. ariketa multzoko 2. ariketa).

Ariketa03

Idatzi burbuilaren ordenaketa metodoa (*Bubble sort*) erabiliz zerrenda bat ordenatzen duen eta ordenatzeko behar duen denbora kalkulatzeko duen programa. Honetarako, lehenik definitu hiru azpiprograma:

- 100 ausazko zenbaki osoen zerrenda bat (1 eta 200 artekoak) sortu eta itzultzen duen azpiprograma.
- Zerrenda bat ordenatuta dagoen itzultzen duen azpiprograma.
- Zerrenda bat eta bi indize emanda, bi posizio horietan dauden elementuak trukatzeko dituen azpiprograma. Honek ez du ezer itzuliko.

Ondoren, jarraitu ondorengo pausoak algoritmoa burutzeko:

- Zerrenda sortu eta erakutsi.
- Gorde aldagai batean uneko ordua *Instant.now()* metodoa erabiliz (*java.time* liburutegia).
- Zerrenda ordenatuta ez dagoen bitartean:
 - Zerrendaren bukaerara ailegatu arte:
 - Bi elementu hartu, konparatu eta lehena bigarrena baino handiagoa bada trukatu.
 - Bestela, indizea handitu hurrengo bi elementuak hartzeko.
 - Indizea hasieratu eta errepikatu.

Puntu honetan zerrenda ordenatuta egongo da. Kalkulatu berriro uneko ordua eta erabili *Duration.between()* iraupena kalkulatzeko eta erakutsi. Erakutsi ordenatutako zerrenda ere.

Funtzionatzen duenean, handitu elementu kopurua 5000 elementutara, egin zenbait exekuzio eta begiratu behar duen denbora.

Ariketa04

Idatzi koktelaren ordenaketa metodoa (*Cocktail shaker sort*) erabiliz zerrenda bat ordenatzen duen eta ordenatzeko behar duen denbora kalkulatzeko duen programa. Berrerabili aurreko ariketan definitutako hiru azpiprogramak.

- Ondoren, jarraitu ondorengo pausoak algoritmoa burutzeko:
- Zerrenda sortu eta erakutsi.
- Gorde aldagai batean uneko ordua *Instant.now()* metodoa erabiliz (*java.time* liburutegia).
 - Zerrenda ordenatuta ez dagoen bitartean:
 - Hasierako eta bukaerako posizioak gorde (hasieran 0 eta luzera – 1 izango dira)
 - Bukaerako posizioa ailegatu arte:
 - Bi elementu hartu, konparatu eta lehena bigarrena baino handiagoa bada trukatu.

- Bestela, indizea handitu hurrengo bi elementuak hartzeko.
- Bukaeran, elementu handiena bukaeran kokatuta egongo da.
- Hasierako posiziora ailegatu arte:
 - Bi elementu hartu, konparatu eta lehena bigarrena baino txikiagoa bada trukatu.
 - Bestela, indizea txikitu hurrengo bi elementuak hartzeko.
 - Bukaeran, elementu txikiena hasierako kokatuta egongo da.
- Hasierako posizioa batean handitu eta bukaerakoa batean txikitu eta errepikatu.

Puntu honetan zerrenda ordenatuta egongo da. Kalkulatu berriro uneko ordua eta erabili *Duration.between()* iraupena kalkulatzeko eta erakutsi. Erakutsi ordenatutako zerrenda ere.

Funtzionatzen duenean, handitu elementu kopurua 5000 elementutara, egin zenbait exekuzio eta begiratu behar duen denbora. Aurreko algoritmoarekin konparatuta zein da azkarragoa?

Ariketa05

Idatzi nahasketa bidezko ordenaketa metodoa (*Merge sort*) erabiliz zerrenda bat ordenatzen duen eta ordenatzeko behar duen denbora kalkulatzeko duen programa. Berrerabili aurreko ariketan definitutako hiru azpiprogramak.