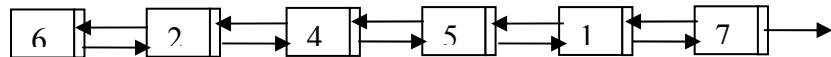


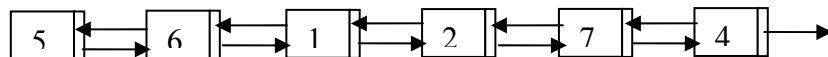
1 ariketa: (3 puntu)

Esteka bikoitzeko zerrenda batean, existitzen diren elementu bikoiti eta bakoitien kopurua **berdina izanda**, lehenabizi elementu bikoiti guztiak kokatuta daude, eta jarraian gainontzeko elementu bakoitiak.
Adibidez:



Diseina eta inplemetatu *tartekatu* metodoa esteka bikoitzeko klase iteratzailean (DoubleLinkedListIter). Metodoak beste zerrenda berri bat sortzen du, jatorrizko nodo bakoiti eta bikoitiak tartekatuz. Hau da, bukaeran jatorrizko zerrenda hutsa izango da, eta sortutako zerrendak nodo guztiak tartekatuta edukiko ditu. Elementuak ez dute derrigorrez jatorrizko zerrendaren ordena mantendu beharrik, baina ematen den ebazpena $O(n)$ izan behar da (zerrendaren elementuak aldi bakar bat atzitu daitezke).

DoubleLinkedList<T> tartekatu() {}



```
class Node<T> {  
    T element;  
    Node<T> next;  
    Node<T> prev;  
}  
public class DoubleLinkedList<T> {  
    Node<T> top;  
    Node<T> actual;  
}
```

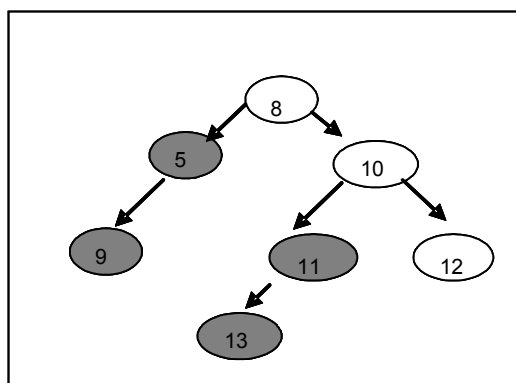
1 ariketa: (3,5 puntu)

Zuhaitz bitar batean, zuhaitz hori ezkertia den ala ez erabakitzen duen metodo errekursiko bat espezifikatu, diseinatu eta inplementatu Javan.

Zuhaitz bitar bat ezkertia da baldin:

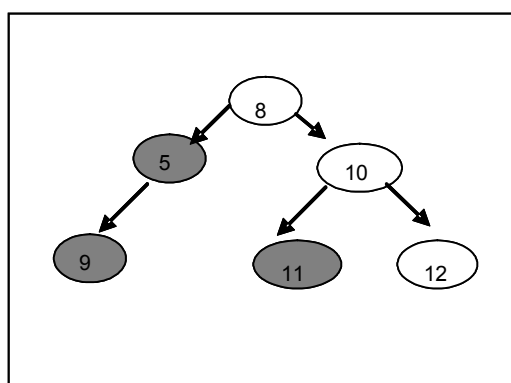
- Hutsa da, edo
- Hostoa da, edo
- Barne-adabegi bakoitzarentzat, ezkerreko ume diren ondorengo guztien kopurua handiagoa da eskuineko ume diren ondorengo guztien kopurua baino.

Adibideak:



Zuhaitza ezkertia da.

- 8: 4 ezkerreko > 2 eskuineko.
- 5: 1 ezkerreko > 0 eskuineko
- 10: 2 ezkerreko > 1 eskuineko
- 11: 1 ezkerreko > 0 eskuineko



Zuhaitza EZ da ezkertia.

- 8: 3 ezkerreko > 2 eskuineko.
- 5: 1 ezkerreko > 0 eskuineko
- 10: 1 ezkerreko EZ > 1 eskuineko

3. ariketa: (4 puntu)

Ikasle guztien irakasgai guztietarako tutoretzak modu orokorrean kudeatzeko Tutoretza orokorrak Klase bat inplementatu nahi da. Honako arauak ezarri dira:

- irakasle bakoitzak gehienez ere 3 irakasgai ematen ditu
- irakasgai bakoitzeko irakasle kopurua, gehienez, 3 da
- irakasle guztiek dute irakasgai bat gutxienez
- irakasgai guztiek dute irakasle bat gutxienez
- ikasle batek tutoretza eskatzen duenean, hurrengo txanda esleitzen zaio. Txanda modu zentralizatuan kudeatzen da: bakarra dago irakasle eta irakasgai guztientzat. Tutoretza irakasgaietarako eskatzen da, ez irakasleetarako.

TutoretzaOrokorrak klasearen hiru eragiketa dira honako hauek:

int hurrengoTxandaEman ()

-- Post: txanda eguneratzen da (bat gehiago), eta txanda eguneratu hori itzultzen du.

void tutoretzaEskatu (String ikaslea, String irakasgaia)

-- Aurre: Irakasgaia existitzen da.

-- Post: ikaslea irakasgaia-ri dagokion tokian eta egokitu zaion txandarekin gordetzen da.

String ikasleariKasuEgin (String irakaslea)

-- Aurre: Irakaslea existitzen da.

-- Post: Irakasleak atenditu behar duen ikaslea lortzen da, eta ikasle hori ezabatu egiten da egituratik.

Irakasleak atenditzen duen ikaslea aukeratzeko irizpidea: irakasleak ematen dituen irakasgaietako bat eskatu dutenen artean, lehenago iritsi dena (txandarik txikiena).

Salbuespena: irakasleak ematen dituen irakasgaietan tutoretza eskatu duen ikaslerik ez badago, *null itzuliko da*

Eskatzen da:

a) **TutoretzaOrokorrak klasea definitu Javan.** Definizio horrek aukera eman behar du espezifikatutako hiru metodoak ordena konstantean inplementatzeko, hau da, hiru algoritmoen **konplexutasun-ordenek independente** izan behar dute tutoretza eskatu duten **ikasleen kopurutik**.

Datu egitura ezagunak erabil daitezke definizioan.

b) *Diseinatu eta inplementatu Javan goian espezifikatutako hiru metodoak, aukeratu duzun egituraren arabera. Justifikatu laburki hiru metodoen konplexutasun-kostua..*