DATU-EGITURAK ETA ALGORITMOAK I 2. MAILA
2005ko ekainaren 27a

1 ariketa (3 puntu)

Elementu osoko esteka sinpleko zerrenda ordenatu batean, diseina eta inplementatu hurrengo metodoa LinkedListItr klasean:

LinkedListItr errepikatuakEzabatu()

Metodoak hurrengoa egiten du:

1. Elementu errepikatuak ezabatzen ditu jatorrizko zerrendatik
2. Nodo errepikatu guztiak beste zerrenda batean itzultzen ditu.

Diseinua

Zerrenda berri bat sortu
Unekoa kokatu lehengo elementuan
Bitartean elementuak zerrendan
    Uneko elementuaren balioa gorde (
    Bitartean uneko elementuaren balioa == hurrengo elementuarena
        1.1 Txertatu zerrenda berrian hurrengo nodoa
    Hurrengo nodora joan
    Eguneratu unekoa
Hurrengo nodora joan
Itzuli zerrenda berria

1.1 Txertatu zerrenda berrian hurrengo nodoa
  If (zerrenda hutsa)
    Nodoa kokatu zerrendaren lehendabiziko elementu bezala
    Unekoa=nodo berria
  Bestela
    Unekoaren hurrengoa erreferenziatu nodo berrira
    Uneko=nodo berria

Inplementazioa

```
public LinkedList errepikatuakEzabatu(){
  LinkedList<T> zBerria=new LinkedList<T>();
  actual=top;
  int lehen;
 while (actual!=null){
  lehen=(actual.getElem()).intValue();
  while ((actual.next!=null) &&(actual.next.elem).intValue()==lehen)){
    zBerria.txertatuNode(actual.next);
        actual.next=actual.next.next;
  }
  actual=actual.next;
 }
 return zBerria;
}
```

```
public void txertatuNode(Node nodo){
   if (isEmpty()){
     setTop(nodo);
     actual=nodo;
   }else {
     actual.next=nodo;
     actual=nodo;
   }
 }
```

2 ariketa (3 puntu)

Zuhaitz bitar bat **GorriBeltza (GB)** da, baldin eta soilik baldin honako baldintzak betetzen baditu:

1) **hosto** guztiak **beltzak** dira,
2) **gorria** den edozein adabegik **bi ume beltz** ditu,
3) edozein adabegi hartuta, bere ondorengo hosto batera doan **edozein bidek** adabegi **beltzen kopuru berdina** du.

Espezifikatu, diseinatu eta inplementatu Java-n zuhaitz bitar iteratzaile klasean hurrengo metodoa:

boolean isGorriBeltza()

egiazkoa izango dena baldin eta soilik baldin zuhaitza gorribeltza bada.

a)Erazagupena

Sarrea: Zuhaitz bitarra.
Irteera: boolean
Post: true bss zuhaitza gorribeltza bada.

Adabegi bakoitzean jakin behar dugu bere azpizuhaitz bakoitzentzako, ea gorribeltzak diren ala ez, eta zenbat adabegi beltz dauzkate. Horretarako murgilkera erabili behako dugu.

isGB,zBeltza = isGorriBeltza(node)

b) Diseinua

Kasu nabariak:

node.isHutsa() → isGB:=true; zBeltza=0
node.isHostoa() → if (node.isBeltza()) -> isGB:=true; zBeltza=1
else isGB:=false; zBeltza=0

Kasu orokorra:

```
(isGBEzk, zBeltzaEzk)=isgorriBeltza (node.ezkerra)
(isGBEsk, zBeltzaEsk)=isgorriBeltza (node.eskubia)
 isGB= isGBEzk eta isGBEsk
 zBeltza=max(zBeltzaEzk, zBeltzaEzk) // baten bat 0 izan daiteke
 if (!node.isBeltza())
   Ezkerreko umea badauka -> isGB=isGB && node.ezkerUmea.isBeltza()
   Eskubiko umea badauka -> isGB=isGB && node.eskuinUmea.isBeltza()
 endIF
 Ezker eta eskubiko umeak badauzka -> isGB=isGB && (zBeltzaEzk= zBeltzaEsk)
 if (node.isBeltza())
    zBeltza= zBeltza+1;
 itzuli (isGB,zBeltza)
```

Inplementazioa

```
private Context isGorriBeltza(BTNode pNode){
if (pNode==null)
  return new Context(0,true);
 else if ( (pNode.getLeft()==null) && (pNode.getRight()==null))  // hostoa da
  if (pNode.isBeltza())
   return new Context(1,true);
   else
    return new Context(0,false);
 else {// ez da ostoa
  Context ctxEzk=isGorriBeltza(pNode.getLeft());
  Context ctxEsk=isGorriBeltza(pNode.getRight());
  boolean isGB=ctxEzk.isGB && ctxEsk.isGB;
  int z=max(ctxEzk.zBeltza,ctxEsk.zBeltza);
  if (!pNode.isBeltza()) {
   z=max(ctxEzk.zBeltza,ctxEsk.zBeltza);
   if (pNode.getLeft()!=null)
      isGB=isGB&&((BTNode)pNode.getLeft()).isBeltza();
   if (pNode.getRight()!=null)
      isGB=isGB&&((BTNode)pNode.getRight()).isBeltza();
  }
  if ((pNode.getLeft()!=null) && (pNode.getRight()!=null))
      isGB=isGB && (ctxEzk.zBeltza==ctxEsk.zBeltza);
  if(pNode.isBeltza()) z++;
  return new Context(z,isGB);
 }
}
```

Beste klase osagarriak

```
public boolean isGorriBeltza(){
 Context ctx=isGorriBeltza(root);
 return ctx.isGB;
}
```

```
private class Context {
  int zBeltza;
  boolean isGB;
  Context(int pBeltza, boolean
isGorriBeltza) {
     zBeltza=pBeltza;
     isGB=isGorriBeltza;
    }
```

3. ariketa: (4 puntu)

a) Diseinatu eta inplementatu Javan **void taulaKargatu(String fitx)** metodoa Sinonimoak klasean. Metodo horrek sinonimoen hash taula osatzen du, fitxategi-izen bat ("sinonimoak.txt") emanda.

Diseinua

Hash taula sortu
Fitxategi irakurlea hasieratu fitxategiarekin
Bitartean lerroak fitxategian
  Gakoa=fitxategiaren lerroa irakurri
  Sinonimoak=fitxategiaren lerro irakurri
  Ilara berri bat sortu
  Bitartean sinonimoak lerroan
        Ilaratu sinonimoa lerroan
  Txertatu hash taulan gakoa, bere ilararen sinonimoekin

Inplementazioa

```
void taulaKargatu(String fitx){
  HashTable<String,Queue<String> ht=new HashTable<String,Queue<String>>();
  FileReader fr=new FileReader(fitx);
  Vector v1,v2;
  String gakoa;
  while(fr.isLerroGehiago){
   v1=fr.getHitzak();
   v2=fr.getHitzak();
   gakoa=v1.elementAt(0);
   Enumeration<String> e=v2.elements();
   String sinonimo;
   Queue<String> q=new Queue<String>();
   while (e.hasMoreElements(){
    sinonimo=e.nextElement();
    q.enqueue(sinonimo);
   }
   ht.insert(gakoa,q);
  }
}
```

b) Diseinatu eta inplementatu Javan **void aldatu(String fitx)** metodoa. Metodo horrek jatorrizko testu bat ("testua.txt") moldatzen du sinonimoen hash taula erabiliz, eta emaitza pantailatik aurkezten du. Moldaketa egiteko testuko hitzen ordez beren sinonimoak jartzen dira, honako arau hauek betez:

- Hitza ez badago hash taulan edo sinonimorik ez badauka, hitza bera idazten da irteera-fitxategian.
- Hitza hash taulan badago (gako moduan) eta sinonimoren bat badauka, sinonimoen ilarako hasierako elementua idatziko da pantailan. Sinonimo hori ez da berriz aukeratuko, harik eta hitz horren gainontzeko sinonimoak erabiltzen diren arte.

Diseinua

Fitxategi irakurlea hasieratu fitxategiarekin
Bitartean lerro gehiago fitxategian
    Lerroa irakurri
     Lerroaren hitz bakoitzentzat
      Hitza hash taulan dagoen begiratu
    ┌ Hitza badago
    │ ┌Sinonimoa badauka
    │ │ Hitzaren sinonimoa lortu bere sinonimoen ilaratik
    │ │ Sinonimoa idatzi pantailan
    │ │ Sinonimoa kokatu ilararen azken elementuan
    │ └Bestela hitz berdina idatzi
    └ Bestela hitz berdina idatzi
endBitartean

Inplementazioa

```
public void aldatu(String fitx) {
  FileReader fr=new FileReader(fitx);
  String sinonimoa,hitza;
  Queue<String> sinonimoak;
  while(fr.isLerroGehiago()){
   lerroa=fr.getHitzak();
   Enumeration<String> e=lerroa.elements();
   while (e.hasMoreElements(){
    hitza=e.nextElement();
    if (ht.exist(hitza))
     sinonimoak=ht.find(hitza);
     if (!sinonimoak.isEmpty()){
      sinonimoa=sinonimoak.front();
      System.out.print(sinonimoa+" ");
      q.enqueue(q.dequeue());
     } else
        System.out.print(hitza+" ");
     else
      System.out.print(hitza+" ");
   }
  System.out.println("");
  }
}
```