

1. **ariketa.** Kalkulatu algoritmo bakoitzaren ordena, matrizearen tamainaren arabera.
Zein da eraginkorrena?

```
type Matrizea is array (1 .. Lerroak, 1 .. Zutabeak) of Integer;
```

```
function Batura1 (Mat :  
                  in Matrizea)  
  return Integer is  
    Guztira : Integer;  
    I, J: Integer;  
begin  
  Guztira:= 0;  
  I := 1;  
  
  while (I <= Lerroak) loop  
    J := 1;  
    while (J <= Zutabeak) loop  
      Guztira:=  
        Guztira + Mat(I, J);  
      J := J + 1;  
    end loop;  
  
    I := I + 1;  
  end loop;  
  
  return Guztira;  
end Batura1;
```

```
function Batura2(Mat: in Matrizea )  
  return Integer is  
    Guztira: Integer;  
    I, J: Integer;  
begin  
  Guztira:= 0;  
  I := 1; J := 1;  
  
  while (I <= Lerroak) loop  
    Guztira:=  
      Guztira + Mat(I, J);  
    if (J < Zutabeak)  
      then J := J + 1;  
      else J := 1;  
      I := I + 1;  
    end if;  
  end loop;  
  
  return Guztira;  
end Batura2;
```

```
function Batura3(Mat :  
                  in Matrizea )  
  return Integer is  
  
    Guztira: Integer;  
  
begin  
  Guztira:= 0;  
  
  for I in 1 .. Lerroak loop  
    Guztira := Guztira +  
      Errenkadaren_Batura (Mat, I);  
  end loop;  
  
  return Guztira;  
end Batura3;
```

```
function Errenkadaren_Batura(  
  Mat : in Matrizea;  
  Errenkada: in Integer)  
  return Integer is  
    Guztira: Integer;  
  
begin  
  Guztira:= 0;  
  for J in 1 .. Zutabeak loop  
    Guztira:= Guztira +  
      Mat(Errenkada, J);  
  end loop;  
  return Guztira;  
end Errenkadaren_Batura;
```

2. ariketa. Honako algoritmoaren konplexutasun-kostua aztertu:

```
function Muga_Gainditzen (n: Integer) return Integer is

    Jauzia: Integer := 1;
    Posizioa: Integer := 0;
begin
    while Posizioa < n loop
        Posizioa := Jauzia;
        Jauzia := jauzia * 2;
    end loop;
    return Posizioa;
end Muga_Gainditzen;
```

3. ariketa. ariketa. Ikasleen zerrenda bat dugu, eta ikasle bakoitzeko matrikulatu den asignaturen zerrenda.

- Ikasle baten zenbakia emanda, zenbat denbora hartuko du ikasle hori algebran matrikulatuta dagoen jakiteak?
- Asignatura bakoitzeko, matrikulan dituen ikasleen zerrenda lortu nahi da. Zein da algoritmo horren kostua?

4. ariketa. Hirugarren kurtsoko ikasle batek algoritmo bat egin du, “primitiva”ko txartelen zerrenda emanda (bakoitza dagokion sariarekin) sari handienak dauzkaten 1.000 txartelak lortzeko.

```
type Txartelen_Zerrenda is record  
    Taula: array (1 .. Max) of Txartela;-- Txartela mota definituta  
dago  
    Zenbat: Integer;  
end record;
```

```
function 1000_Hoberenak (Txartelak : in Txartelen_Zerrenda)  
    return Txartelen_Zerrenda is  
-- post: Txartelak zerrendako 1.000 hoberenak lortu dira emaitzan  
  
    Emaidza: Txartelen_Zerrenda;  
begin  
    Hasieratu(Emaidza);  
    I:= 1;  
  
    while (I <= Txartelak.Zenbat) loop  
        if Emaidza.Zenbat < 1000 or  
            Saria(Txartelak.Taula(I)) > Saria(Txikiena(Emaidza))  
        then sartu txartel hau Emaidza-ren bukaeran  
            Ordenatu(Emaidza); -- burbuilaren bidez  
        end if;  
    end loop;  
  
    return Emaidza;  
end 1000_Hoberenak;
```

Ondokoa eskatzen da:

- a) Aztertu soluzioa eta bere kostua kalkulatu
- b) Algoritmo hori 14.000.000 txartelen zerrenda batean aplikatzen denean, ikasleak dio 8/9 ordu hartzen dituela, 1.000 txartel hoberenak lortzeko. Lagunduko zenioke algoritmoa hobetzen? Zure erantzuna arrazoitu.

5. ariketa Klase hau emanda:

```
public class Liburu {  
    int mota;  
    // 3 balio posible: 1 (abenturak), 2 (historia) edo 3 (zientzia)  
    String izenburua;  
    String egileak;  
}
```

Liburuen array bat ordenatu nahi da, lehenengo abenturazkoak egoteko, ondoren historiakoak eta azkenean zientziakoak. Katgoria bakoitzaren barruan, ordenak ez du garrantzirik. 2 algoritmo eman dizkigute:

Algoritmo 1

```
public void bubbleSort(Liburu[] taula) {  
    int out, in;  
    for (out = taula.length - 1; out > 0; out--)  
        for (in = 0; in < out; in++)  
            if (taula[in].mota > taula[in + 1].mota)  
                swap(taula, in, in + 1);  
}
```

Algoritmo 2

```
public void misterio(Liburu[] taula) {  
    int in, i1, i2, i3 = 0;  
    Liburu[] t1, t2, t3; // 3 array lagungarri  
    t1 = new Liburu[taula.length];  
    t2 = new Liburu[taula.length];  
    t3 = new Liburu[taula.length];  
  
    for (in = 0; in < taula.length; in++)  
        if ( taula[in].mota == 1 ) { t1[i1] = taula[in]; i1++; }  
        else if ( taula[in].mota == 2 ){ t2[i2] = taula[in]; i2++; }  
        else if ( taula[in].mota == 3 ){ t3[i3] = taula[in]; i3++; }  
    }  
    int i = 0;  
    for (in = 0; in < i1; in++) { taula[i] = t1[in]; i++;}  
    for (in = 0; in < i2; in++) { taula[i] = t2[in]; i++;}  
    for (in = 0; in < i3; in++) { taula[i] = t3[in]; i++;}  
}
```

Hau eskatzen da:

- Azaldu zer egiten duen algoritmo bakoitzak
- Esan, modu arrazoituan, zein den algoritmo bakoitzaren kostua, eta horren arabera, 2 algoritmoetatik zein izango litzatekeen eraginkorrena

6. ariketa Bi algoritmo hauek emanda:

Algoritmo 1

```
public static int count(int[] a) {  
    // a[i] guztiak != 0 eta desberdinak  
    int N = a.length;  
    int cnt = 0;  
    for (int i = 0; i < N; i++) {  
        for (int j = i+1; j < N; j++) {  
            if (a[i] + a[j] == 0) {  
                cnt++;  
            }  
        }  
    }  
    return cnt;  
}
```

Algoritmo 2

```
public static int count(int[] a) {  
    // a[i] guztiak != 0 eta desberdinak  
    int N = a.length;  
    Arrays.sort(a); // mergesort  
    int cnt = 0;  
    for (int i = 0; i < N; i++) {  
        int j = Arrays.binarySearch(a, -a[i]); // bilaketa bitarra  
        if (j > i) cnt++;  
    }  
    return cnt;  
}  
  
public int binarySearch(int[] a, int x)  
// Aurrebaldintza:  
// Postbaldintza: x duenelementuaren posizioa bueltatzen du  
    0 bueltatuko da x ez badago a arrayan
```

Hu eskatzen da:

- a) Esan zer egiten duen algoritmo bakoitzak
- b) Beraien kostua kalkulatu era arrazoituan

Balioak ordenatu (0,5 puntu)

Ikasle batek algoritmo hau idatzi du zerrenda bateko balioak ordenatzeko:

```
public ArrayList<String> ordenatu(ArrayList<String> l) {  
    // post: emaitza sarrerako zerrenda izango da, elementuak  
    //    ordenatuta dituelarik  
  
    ArrayList<String> emaitza = new ArrayList<String>();  
  
    Iterator<String> it = l.iterator();  
  
    while it.hasNext(){  
        String s = it.next();  
        emaitza.addLast(s);  
        emaitza.quicksort();  
    }  
  
    return emaitza;  
}
```

a) Kalkulatu algoritmo horren kostua, **modu arrazoituan**.

b) Esan ea problema hori ebazteko soluzio eraginkorrago bat ikusten duzun. Baiezko kasuan, idatzi soluzio berri horren kodea eta esan zein den bere kostua.

Bi zenbaki positiboak emanda x eta n , esan zer kalklatzen duen ondoko metodoak. Zein da bere kostua? Erantzuna arrazoituz.

```
public static Integer exponencial(Integer x, Integer n)
{
    Integer resultado; Integer exp;

    if n == 0 return 1;
    else { resultado = x;
          exp = 1;

          while (2 * exp < n)
              resultado = resultado * resultado;
              exp = 2 * exp;
          }
    return resultado
}
```

Listari buelta eman

Bi algoritmo ditugu zerrenda bateko elementuei buelta emateko:

a)

```
public ArrayList<T> invertir(ArrayList<T> l) {  
    // post: el resultado contiene los elementos de la lista original en  
    // orden inverso  
    ArrayList<T> resultado = new ArrayList<T>();  
    Iterator<T> it = l.iterator();  
    while it.hasNext(){  
        T s = it.next();  
        resultado.addFirst(s);  
    }  
    return resultado;  
}
```

b)

```
public ArrayList<T> invertir(ArrayList<T> l) {  
    // post: el resultado contiene los elementos de la lista original en  
    // orden inverso  
    ArrayList<T> resultado = new ArrayList<T>();  
    Stack<T> aux = new Stack<T>();  
  
    Iterator<T> it = l.iterator();  
  
    while it.hasNext(){  
        T s = it.next();  
        aux.push(s);  
    }  
  
    while (!aux.isEmpty()){  
        T s = aux.pop();  
        resultado.addLast(s);  
    }  
  
    return resultado;  
}
```

a) Kalkulatu algoritmoen kostua, **modu arrazoituan**

b) Esan zein soluzio den eraginkorrena