

**Problema.** Primos entre 1 y n mediante la Criba de Eratóstenes.

**Algoritmo**

```
//inicializar criba con todos los numeros
criba={2,3,...,n}; primos={1};
//repetir hasta que criba esté vacía
while( criba ≠ {} ){
    //agregar menor de la criba a los primos
    m = min(criba);
    primos = primos ∪ {m};
    //eliminar menor y sus múltiplos de la criba
    for(int i=m; i<=n; i+=m)
        if( i ∈ criba )
            criba = criba - {i};
}
escribir primos
```

Ejecución  
¿n=20?

m	primos	criba
	{1}	{2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20}
2	{1,2}	{3,5,7,9,11,13,15,17,19}
3	{1,2,3}	{5,7,11,13,17,19}
5	{1,2,3,5}	{7,11,13,17,19}
7	{1,2,3,5,7}	{11,13,17,19}
11	{1,2,3,5,7,11}	{13,17,19}
13	{1,2,3,5,7,11,13}	{17,19}
17	{1,2,3,5,7,11,13,17}	{19}
19	{1,2,3,5,7,11,13,17,19}	{}

**Programa.** Usando clase Conjunto que permite operar con conjuntos de enteros

```
static public void main(String[]args)
throws IOException
{
    int n = U.readInt("Primos entre 1 y ?");

    Conjunto
        vacio = new Conjunto(),    //{ }

        primos = new Conjunto(1),  //{1}

        criba = new Conjunto(2,n); //{2,3,...,n}
```

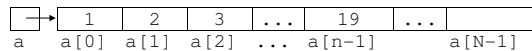
```
//while( criba ≠ {} )
while(! criba.equals(vacio))
{
    //m=menor(criba)
    int m=criba.menor();

    //primos = primos ∪ {m};
    primos=primos.union(new Conjunto(m));

    for(int i=m; i<=n; i+=m)
        //if( i ∈ criba ) criba=criba-{i};
        if( criba.contiene(i) )
            criba=criba.resta(new Conjunto(i));
}
U.println(primos.toString());
}
```

representación?

Ejemplo: {1,2,3,5,7,11,13,17,19}



class Conjunto

```
{
    //variable global para todos los conjuntos
    //(espacio de memoria se reserva una vez)
    static protected final int N=1000;

    //arreglo para elementos
    protected int[]a=new int[N];

    //variable para cardinal (Nº de elementos)
    protected int n;
```

```
//constructores
public Conjunto() {
    n=0;
}
public Conjunto(int x){
    n=1;
    a[0]=x;
}
public Conjunto(Conjunto x){
    for(n=0;n<x.n;++n)
        a[n]=x.a[n];
}
```

## Clase 16:

```
public Conjunto(int x,int y)
{
    n=0;
    for(int i=x; i<=y; ++i)
        agregar(i);
}
protected void agregar(int x)
{
    if(n>=N)
        U.abortar("máximo elementos="+N);

    a[n++]=x; //a[n]=x; ++n;
}
```

```
public boolean contiene(int x)
{
    for(int i=0; i<n; ++i)
        if(a[i]==x) return true;
    return false;
}

public Conjunto union(Conjunto x)
{
    Conjunto aux=new Conjunto(this);

    for(int i=0; i<x.n; ++i)
        if(!aux.contiene(x.a[i]))
            aux.agregar(x.a[i]);

    return aux;
}
```

```
public Conjunto inter(Conjunto x)
{
    Conjunto aux=new Conjunto();

    for(int i=0; i<n; ++i)
        if(x.contiene(a[i]))
            aux.agregar(a[i]);

    return aux;
}
```

```
public Conjunto resta(Conjunto x)
{
    Conjunto aux=new Conjunto();

    for(int i=0; i<n; ++i)
        if(!x.contiene(a[i]))
            aux.agregar(a[i]);

    return aux;
}
```

```
public boolean equals(Conjunto x){
    if(n!=x.n) return false;
    for(int i=0; i<n; ++i)
        if(!x.contiene(a[i])) return false;
    return true;
}
public int cardinal(){
    return n;
}
public int menor(){
    int min = a[0];
    for(int i=1; i<n; ++i)
        min=Math.min(a[i],min);
    return min;
}
```

```
//entregar "{n¹,n²,...,n²}"
public String toString(){
    String aux="{ ";
    for(int i=0; i<n; ++i)
        aux += a[i] + (i<n-1 ? ", " : "");
    return aux + "}";
}
public String toString(){
    if(n==0) return "{}";
    String aux = "{ " + a[0];
    for(int i=1; i<n; ++i)
        aux += ", " + a[i];
    return aux + "}";
}

} //fin class Conjunto
```

**Problema.** Reescriba la clase Conjunto para la siguiente representación para conjuntos de enteros entre 0 y N-1

Ejemplo: {1,2,3,5,7,11,13,17,19}

	→	false	true	true	true	false	...	
a		a[0]	a[1]	a[2]	a[3]	a[4]		A[N-1]

```
class Conjunto{
    static protected final int N=1000;

    protected boolean[]a=new boolean[N];
    //se inicializa automaticamente con false
}
```

```
//constructores
public Conjunto() {
}
public Conjunto(int x){
    a[x]=true;
}
public Conjunto(Conjunto x){
    for(int i=0;i<N;++i)
        a[i]=x.a[i];
}
public Conjunto(int x,int y){
    for(int i=x; i<=y; ++i)
        a[i]=true;
}
```

```
public Conjunto union(Conjunto x)
{
    Conjunto aux = new Conjunto();

    for(int i=0; i<N; ++i)
        aux.a[i] = a[i] || x.a[i];

    return aux;
}
public Conjunto inter(Conjunto x)
{
    Conjunto aux = new Conjunto();
    for(int i=0; i<N; ++i)
        aux.a[i] = a[i] && x.a[i];
    return aux;
}
```

```
public Conjunto resta(Conjunto x){
    Conjunto aux = new Conjunto();
    for(int i=0; i<N; ++i)
        aux.a[i] = a[i] && !x.a[i];
    return aux;
}
public boolean equals(Conjunto x){
    for(int i=0; i<N; ++i)
        if(a[i] != x.a[i]) return false;
    return true;
}
public boolean contiene(int x){
    return a[x];
}
```

```
public int cardinal(){
    int n=0;
    for(int i=0;i<N;++i)
        if(a[i]) ++n;
    return n;
}
public int menor(){
    for(int i=0; i<N; ++i)
        if( a[i] ) return i;
    return -1;
}
```

```
public String toString()
{
    String aux="{ ";

    for(int i=0; i<N; ++i)
        if(a[i])
            if(aux.equals("{"))//primer elemento?
                aux += i ;
            else
                aux += ", " + i;

    return aux + " }";
}
}
```