Cálculo de números primos con la criba de Eratóstenes

Fernando Bobillo

Enunciado

- Calcular todos los números primos menores que un cierto n

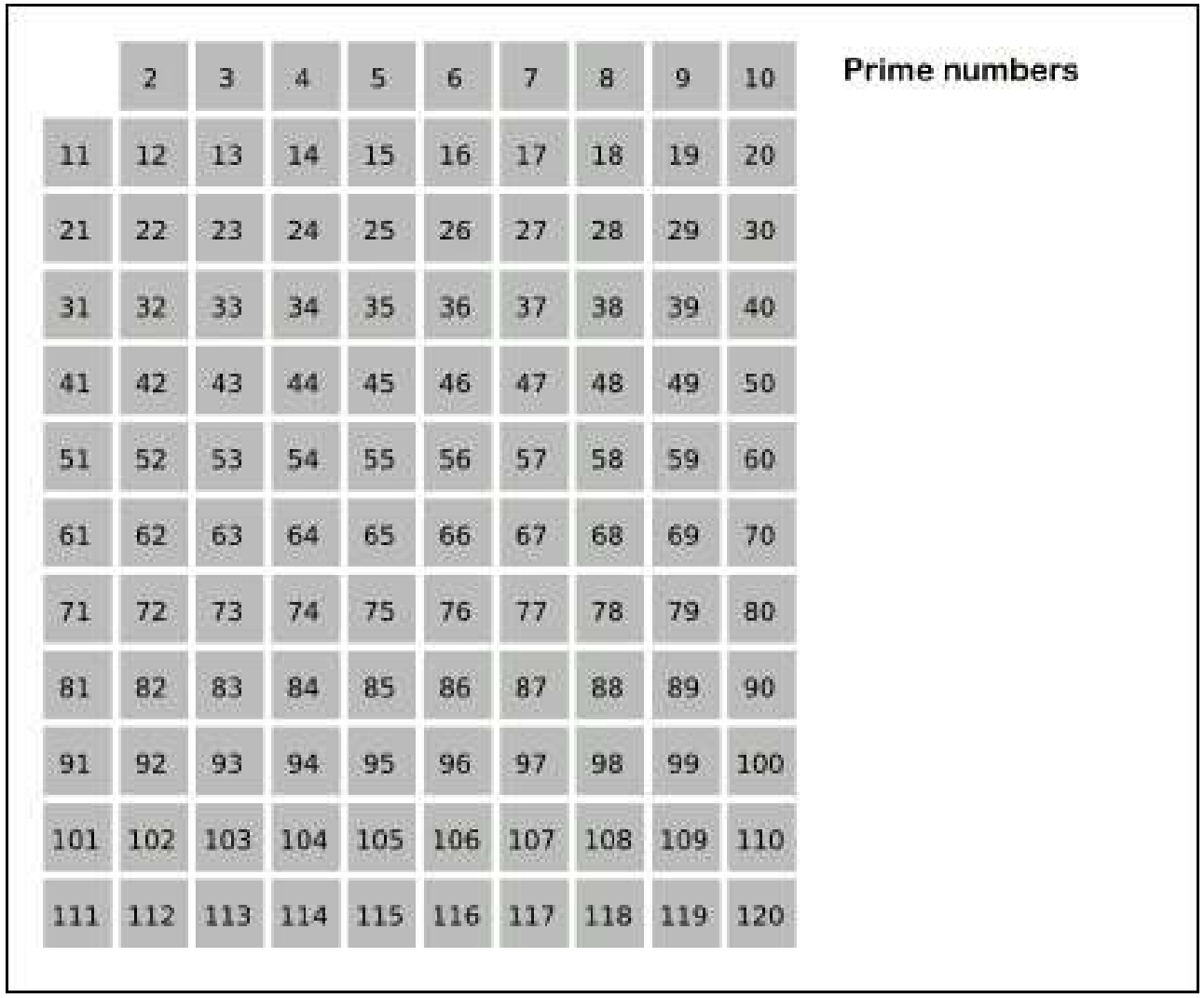
Primos y divisores

- Un número primo es un número natural mayor que 1 cuyos únicos divisores son él mismo y la unidad
- *x* tiene como divisor a *y* si el resto de dividir *x* entre *y* es 0
- Sea un número natural n
  - n es primo si sus únicos divisores son 1 y n
  - n no es primo si tiene algún divisor en $\mathbb{N} \cap [2, n - 1]$
- Teorema fundamental de la aritmética: todo entero positivo mayor que 1 o es primo o es un producto de números primos
  - n no es primo si tiene algún divisor primo en $\mathbb{N} \cap [2, n - 1]$

Criba de Eratóstenes

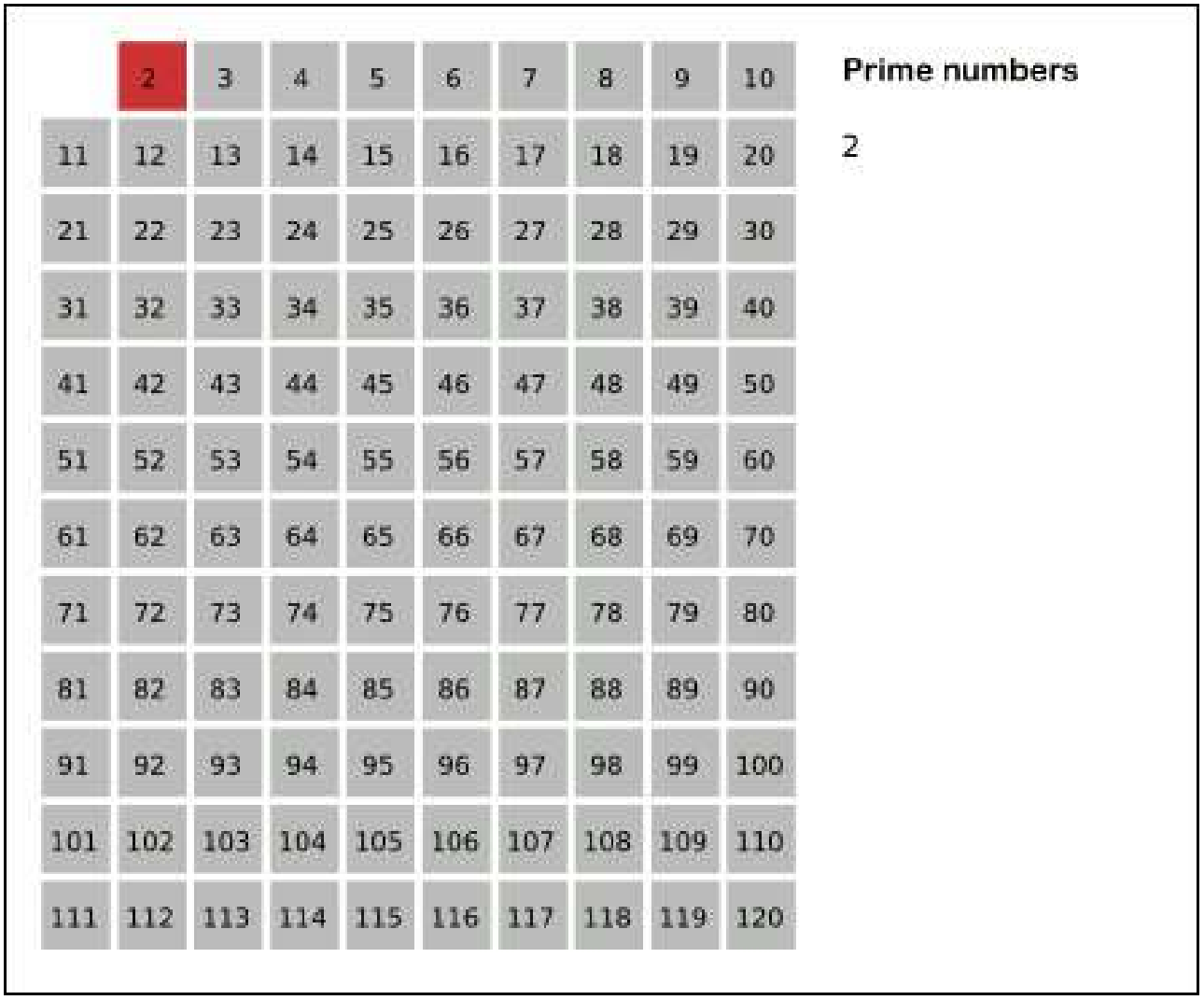
- Algoritmo que halla los n^{os} primos menores que un cierto n
- Idea: mantener una lista de números candidatos a ser primos en [2, n] y tacharlos cuando les encontremos un divisor
- Comenzamos asumiendo que el 2 es un número primo
- Tachamos de la lista todos los múltiplos de 2, pues al tener un divisor (2) no pueden ser primos: 4, 6, 8, 10, 12, …
- El siguiente nº sin tachar de la lista (3) es primo también
- Tachamos de la lista todos los múltiplos de este número (3)
- Repetimos el proceso hasta recorrer toda la lista
- Los números sin tachar de la lista son los primos buscados

Ejemplo de la criba

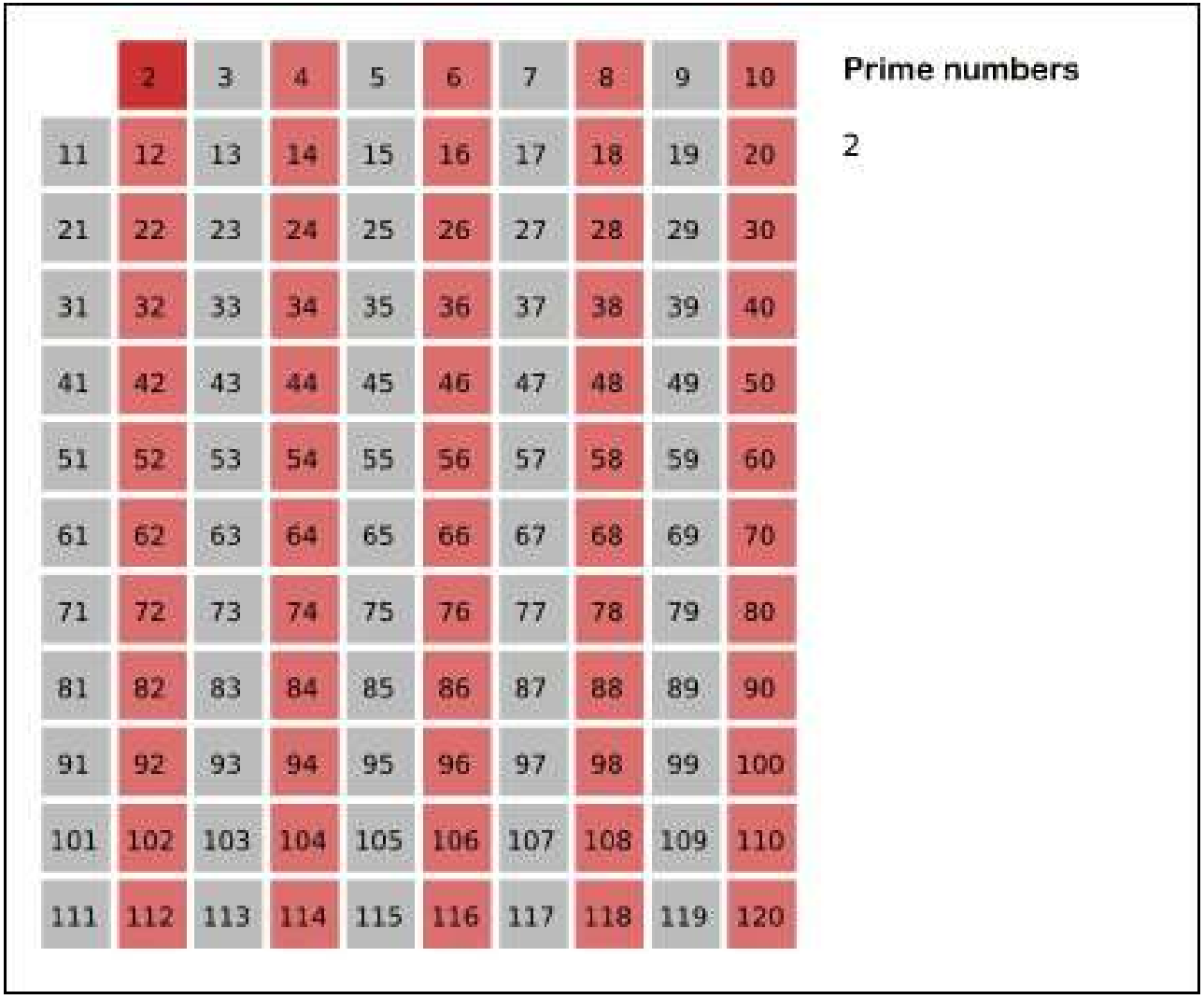


	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120

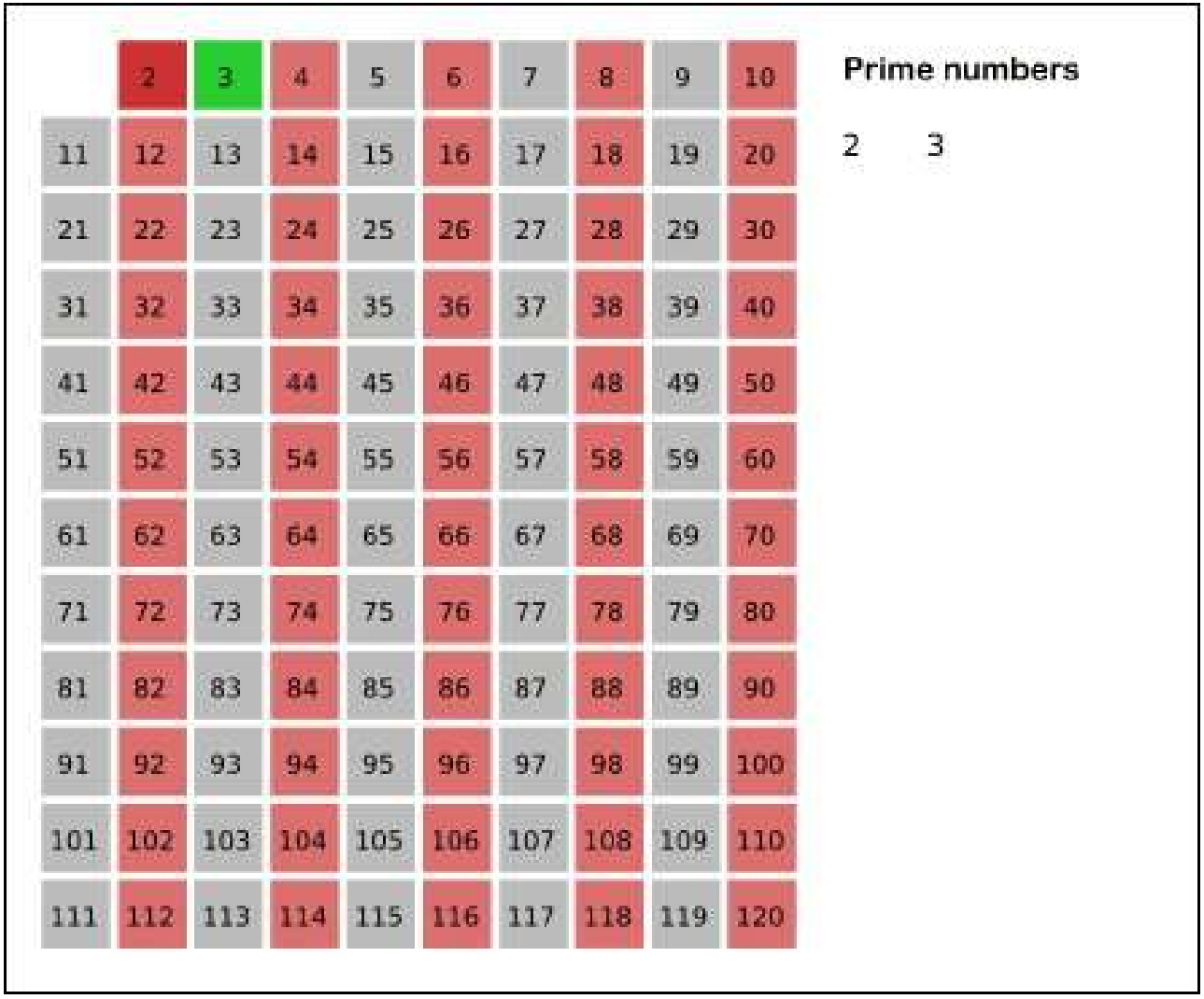
Ejemplo de la criba



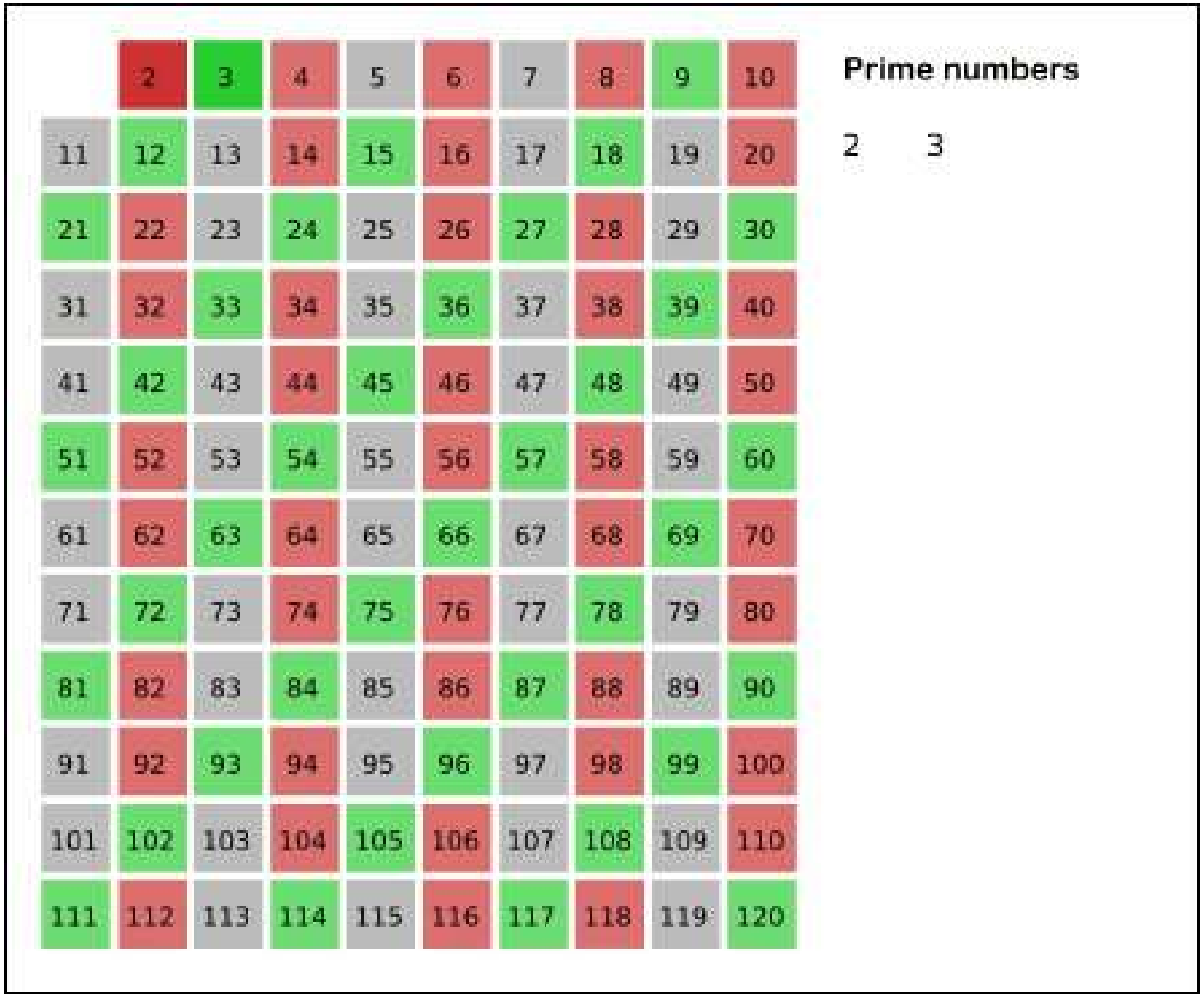
Ejemplo de la criba



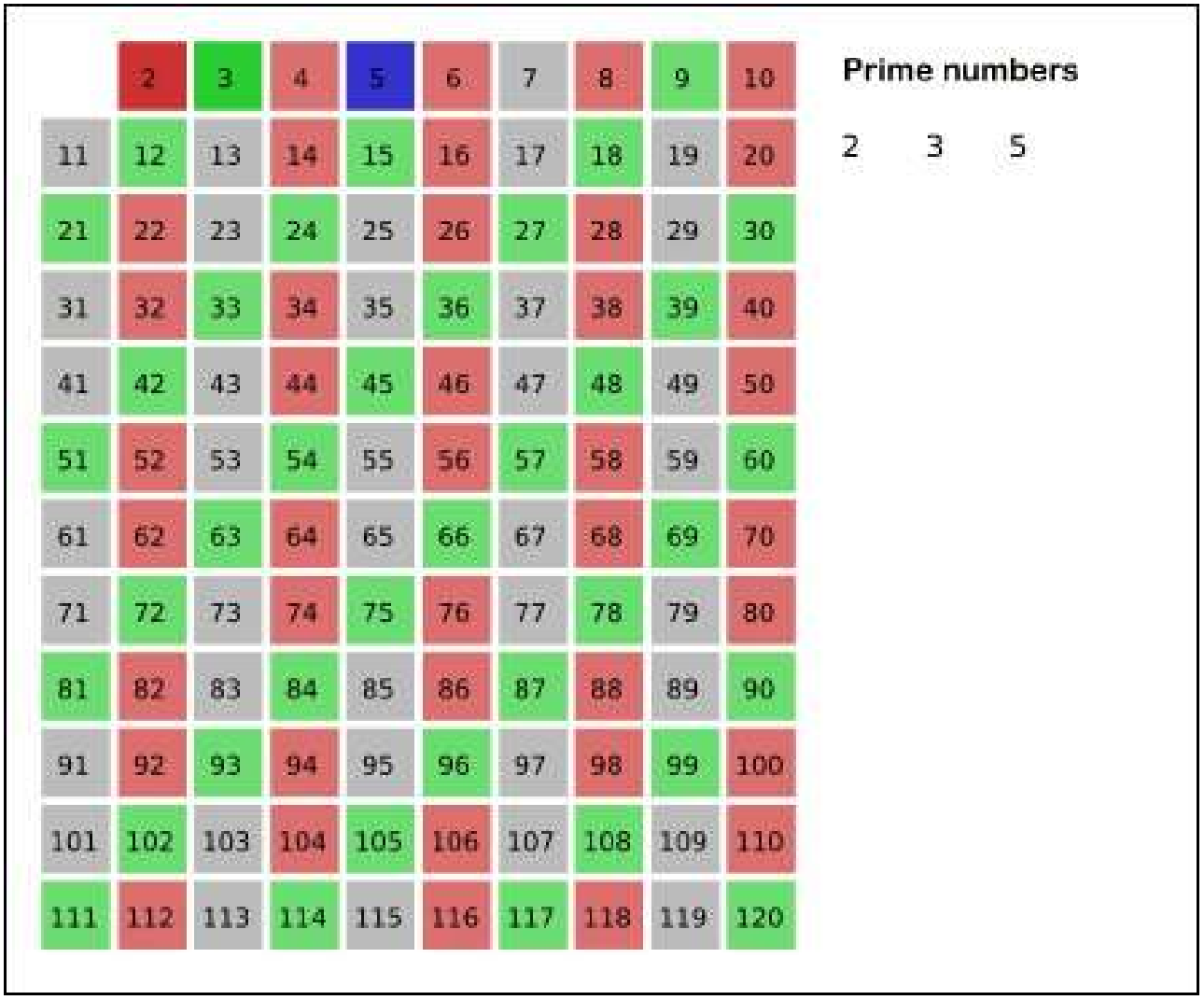
Ejemplo de la criba



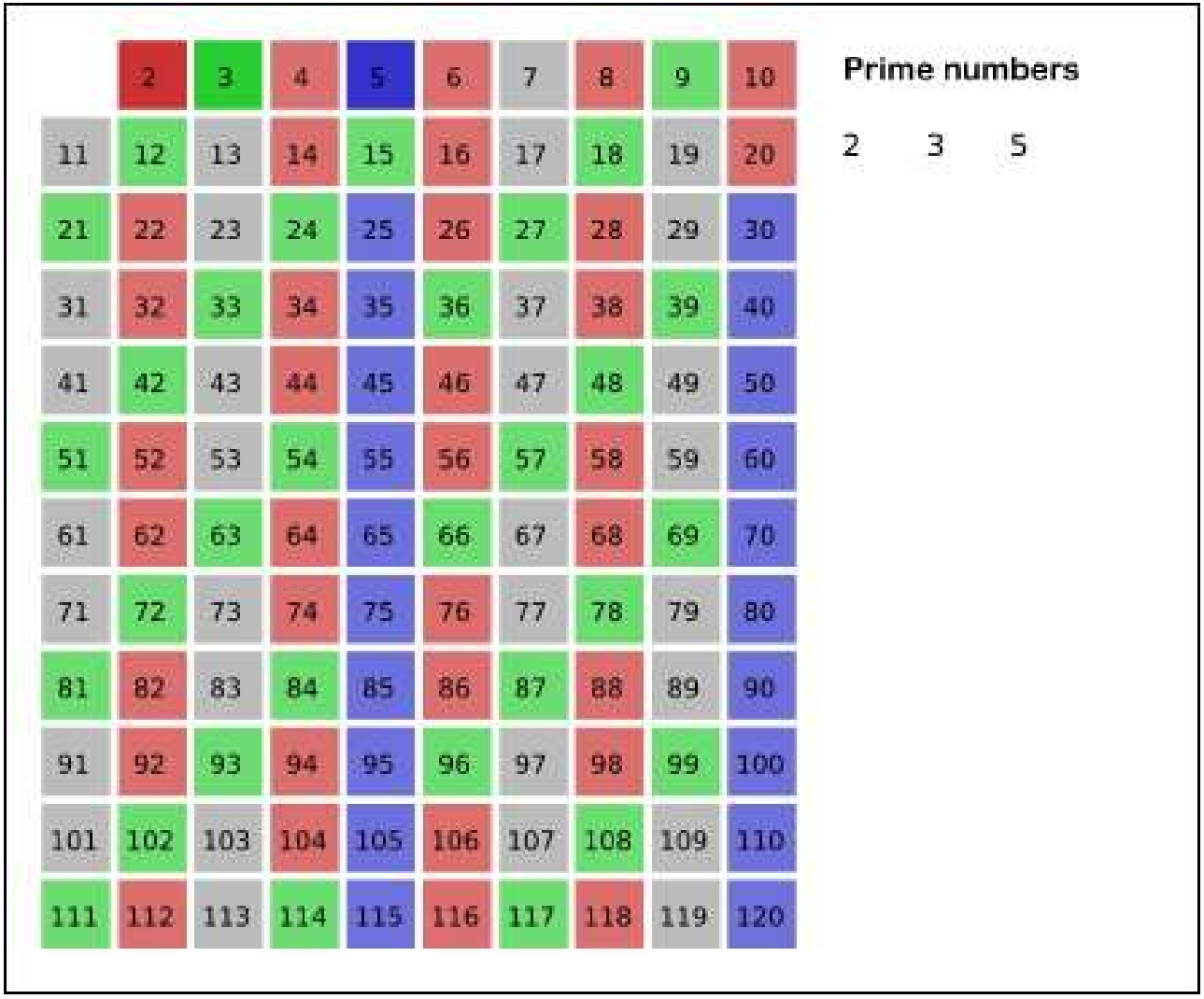
Ejemplo de la criba



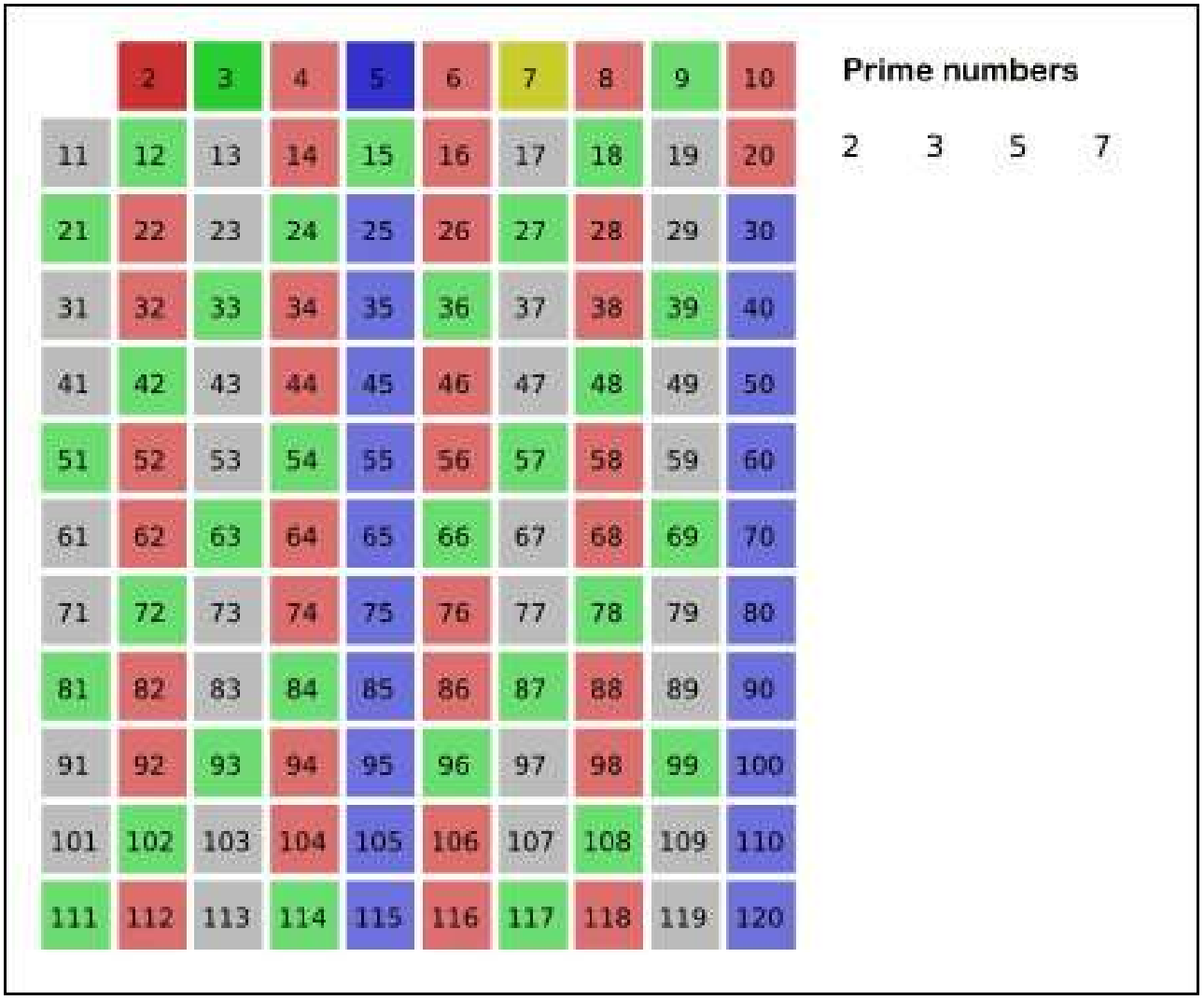
Ejemplo de la criba



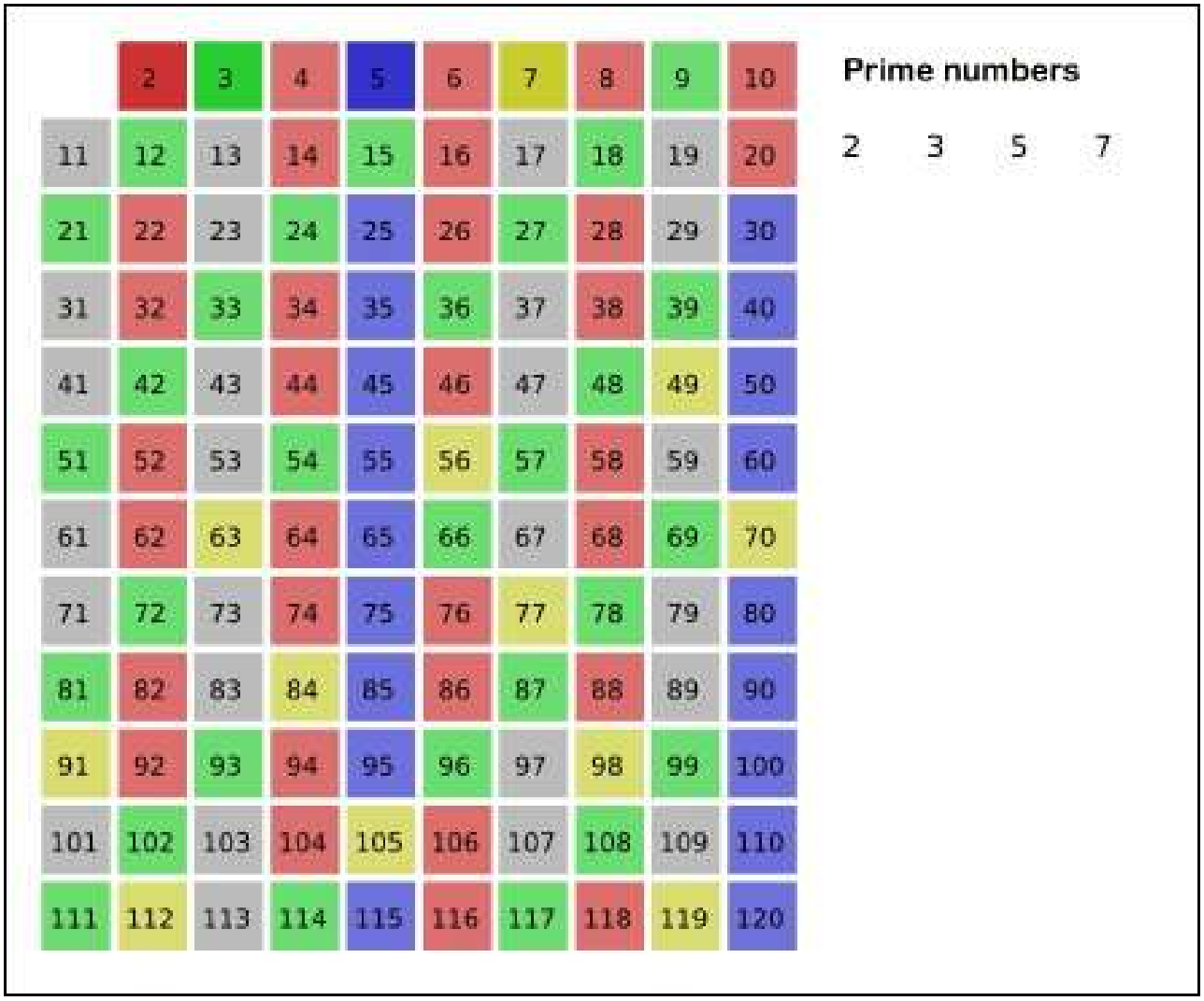
Ejemplo de la criba



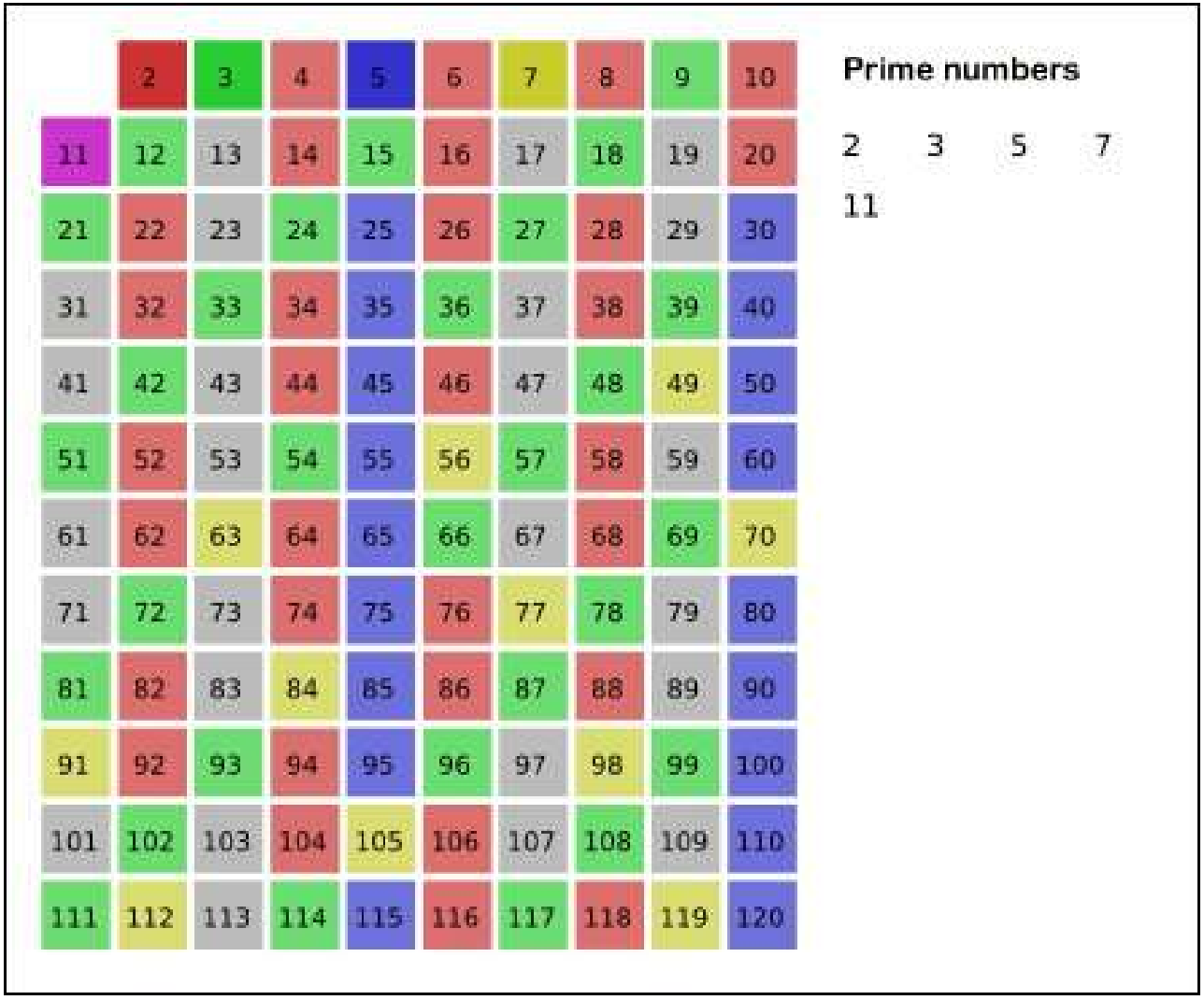
Ejemplo de la criba



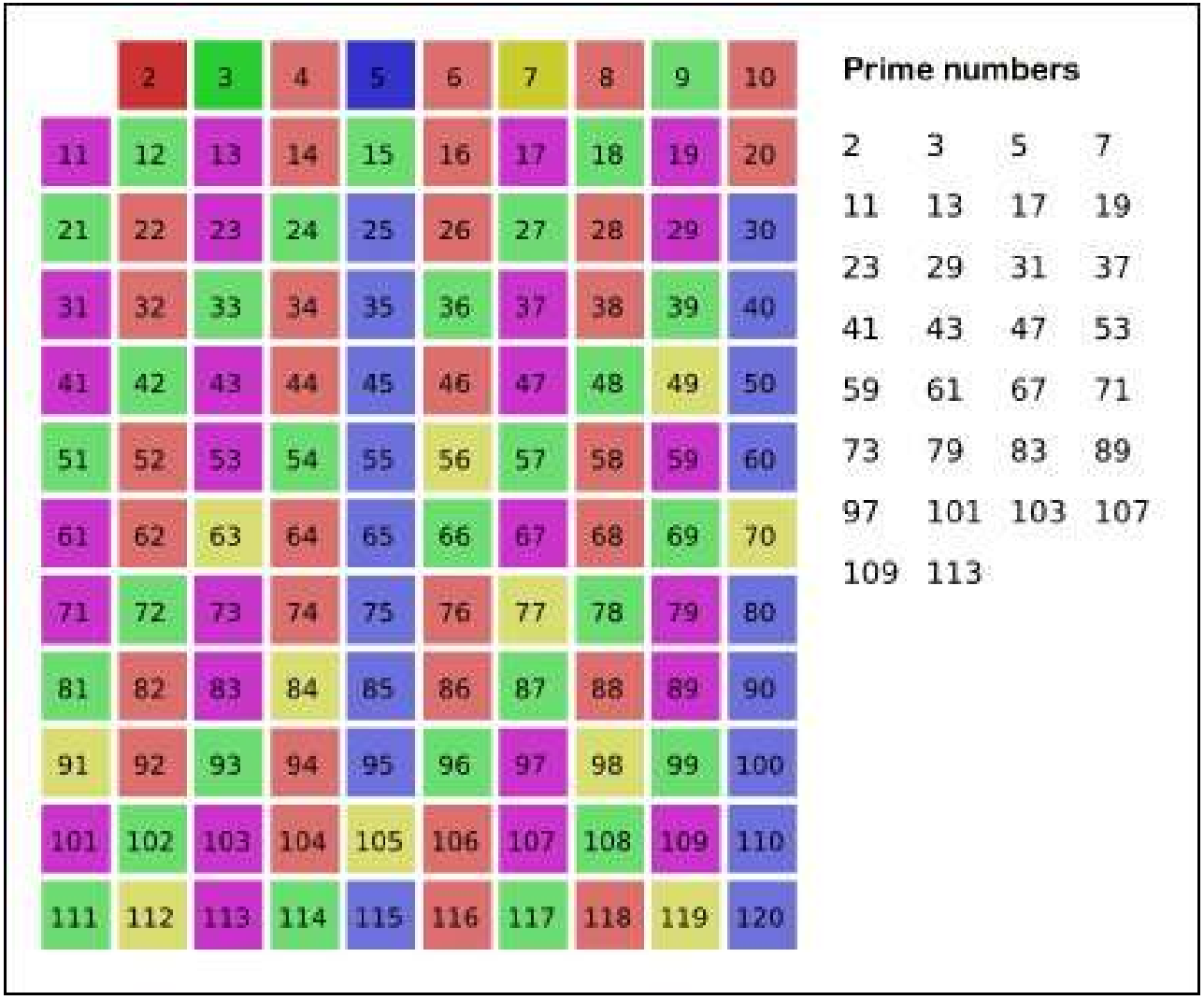
Ejemplo de la criba



Ejemplo de la criba

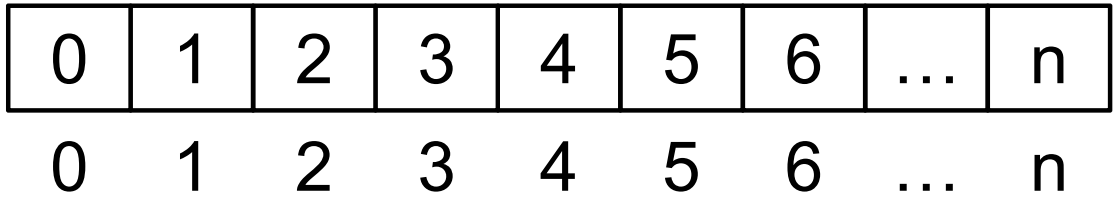


Ejemplo de la criba



Solución 1

- Utiliza un vector `numero` de n + 1 números enteros
  - En realidad las posiciones 0 y 1 no se utilizan
- `numero[i] = i` indica que el número no está tachado
  - El número i todavía es un candidato a ser primo
- `numero[i] = 0` indica que el número ha sido tachado
  - Definitivamente i no es un número primo



- En Java, cada entero ocupa 4 B
  - Por tanto, el vector ocupa 4 (n + 1) B

Código 1

```
Scanner sc = new Scanner(System.in);
System.out.print("Introduce un numero entero: ");
int n = sc.nextInt();
int[] numero = new int[n + 1];
for (int i = 0; i <= n; i++)
    numero [i] = i;
int salto;
int j = 0;
for (int i = 2; i <= n; i++) {
    if (numero[i] != 0) {
          salto = i;
          j = i + salto;
          while (j <= n) {
                    numero[j] = 0;
                    j = j + salto;
          }
    }
}
int contadorPrimos = 0;
for (int i = 2; i <= n; i++) {
    if (numero [i] != 0) {
          contadorPrimos = contadorPrimos + 1;
          System.out.print(i + " es primo \n");
    }
}
System.out.print("\nTotal: " + contadorPrimos + " números primos\n");
```

Solución 2

- Lema: Si n ∈ ℕ+ admite la factorización n = a*b, con a, b ∈ ℤ,

  $a \leq \sqrt{n}$ o $b \leq \sqrt{n}$

- Corolarios:
  - Si n no es primo, uno de sus factores d cumple $1 \leq d \leq \sqrt{n}$
  - Si n no tiene factores d con $1 \leq d \leq \sqrt{n}$, entonces es primo
- n es primo si y solo si es divisible por algún primo $\leq \sqrt{n}$
- Solución 2: considera posiciones del vector con índice $\leq \sqrt{n}$
- El vector sigue ocupando 4 (n + 1) B

Código 2

```
Scanner sc = new Scanner(System.in);
System.out.print("Introduce un numero entero: ");
int n = sc.nextInt();
int[] numero = new int[n + 1];
for (int i = 0; i <= n; i++)
    numero [i] = i;
int salto;
int j = 0;
for (int i = 2; i*i <= n; i++) {
    if (numero[i] != 0) {
          salto = i;
          j = i + salto;
          while (j <= n) {
                    numero[j] = 0;
                    j = j + salto;
          }
    }
}
int contadorPrimos = 0;
for (int i = 2; i <= n; i++) {
    if (numero [i] != 0) {
          contadorPrimos = contadorPrimos + 1;
          System.out.print(i + " es primo \n");
    }
}
System.out.print("\nTotal: " + contadorPrimos + " números primos\n");
```

Solución 3

- Considerar un vector de booleanos en vez de uno de enteros
- `numero[i] = true` indica que el número no está tachado
  - El número i todavía es un candidato a ser primo
- `numero[i] = false` indica que el número ha sido tachado
  - Definitivamente i no es un número primo
- En Java cada booleano ocupa 1 B, el vector ocupa (n + 1) B
  - Reducción del 75%

Código 3

```
Scanner sc = new Scanner(System.in);
System.out.print("Introduce un numero entero: ");
int n = sc.nextInt();
boolean[] numero = new boolean[n + 1];
for (int i = 0; i <= n; i++)
    numero [i] = true;
int salto;
int j = 0;
for (int i = 2; i*i <= n; i++) {
    if (numero[i] != false) {
          salto = i;
          j = i + salto;
          while (j <= n) {
                    numero[j] = false;
                    j = j + salto;
          }
    }
}
int contadorPrimos = 0;
for (int i = 2; i <= n; i++) {
    if (numero [i] != false) {
          contadorPrimos = contadorPrimos + 1;
          System.out.print(i + " es primo \n");
    }
}
System.out.print("\nTotal: " + contadorPrimos + " números primos\n");
```

Solución 4

- Los pares no son primos (salvo el 2): analizar solo los impares

  $\{ 2i + 3 : i = 0, 1, \ldots \lfloor (n-3)/2 \rfloor \} = \{3, 5, 7, 9, \ldots, n / n - 1\}$

  - El vector tendrá únicamente estas $\lfloor (n-3)/2 \rfloor + 1$ posiciones

- Por cada primo $p = 2i + 3$ (tal que $p^2 < n$), debemos eliminar los múltiplos impares de p menores que n, a saber

  $$(2k + 1)\, p = (2k + 1)\,(2i + 3)$$

  para $k = i + 1, i + 2, \ldots$

- El vector ocupa $\lfloor (n-3)/2 \rfloor + 1$ B
  - Para n = 100, la solución 1 ocupa 404 B
  - Para n = 100, la solución 4 ocupa 49 B (12 %)

Código 4

```
Scanner sc = new Scanner(System.in);
System.out.print("Introduce un numero entero: ");
int n = sc.nextInt();
int max = ...;
boolean[] numero = new boolean[max + 1];
for (int i = 0; i <= max; i++)
   numero [i] = true;
for (int i = 0; ... <= n; i++) {
   int k = ...;
   while (... <= n) {
        numero [...] = false;
        k++;
   }
}
int contadorPrimos = 1;
System.out.print("2 es primo \n");
for (int i = 0; i <= max; i++) {
   if (numero [i] != false) {
        contadorPrimos = contadorPrimos + 1;
        System.out.print(... + " es primo \n");
   }
}
System.out.print("\nTotal: " + contadorPrimos + " números primos\n");
```

Código completo en Java

¡TU TURNO!