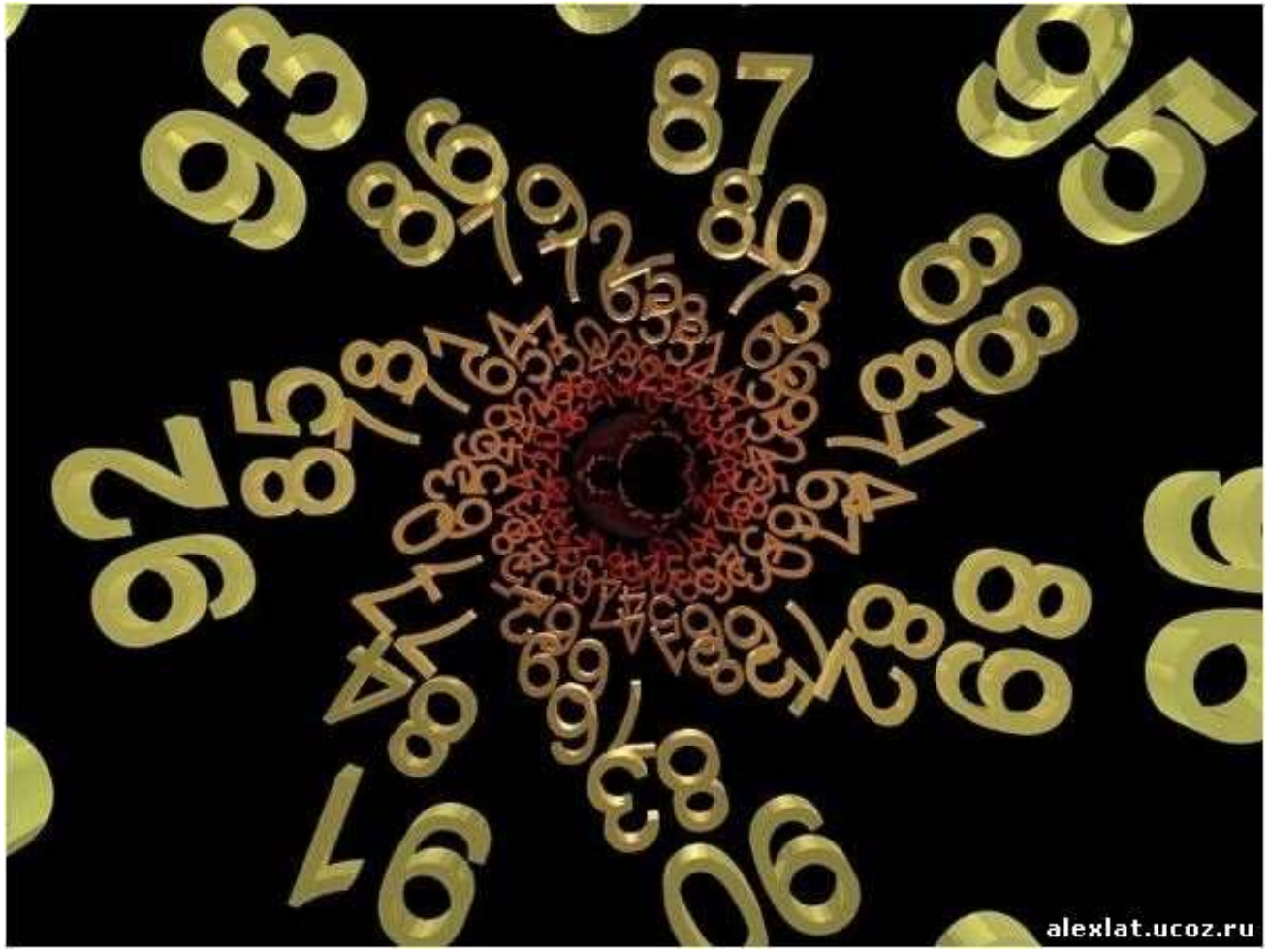
Cálculo del máximo

Fernando Bobillo

Enunciado

- Calcular el máximo de una lista de números naturales



Solución 1

- Dada una lista finita C de números naturales, queremos calcular el más grande
- Podemos asumir que dicha lista es no vacía y que sus elementos están numerados como $c_0, c_1, \ldots, c_{n-1}$
- Se pide encontrar m tal que para todo $x \in C$ se cumple $x \leq m$
- Comenzamos asumiendo que el primer nº (c_0) es el máximo
- Se recorre el resto de la lista y se compara cada valor con el valor del máximo número encontrado hasta ese momento
- Si un elemento es mayor que el máximo, pasa a ser máximo
  - Es decir, se asigna su valor al máximo hasta el momento
- Cuando se termina de recorrer la lista, el máximo número que se ha encontrado es el máximo de toda la secuencia

Pseudocódigo 1

```
programa Maximo1
{ Máximo de una lista no vacía }
definición de variables
  enteros: i, max, n;
  vector de enteros: c;
principio del algoritmo
  lee(n);
  para (i = 0; hasta n - 1; de uno en uno)
    lee(c[i]);
  max = c[0];
  para (i = 1; hasta n - 1; de uno en uno)
    si (c[i] > max)
      max = c[i];
  escribe(max);
fin del algoritmo
```

¿Y si C puede estar vacía?

Solución 2

- Comenzamos asumiendo que el máximo toma valor -1
  - Si C es no vacía, todo natural en C será mayor que -1
- Se recorre la lista desde el inicio y se compara cada valor con el máximo número encontrado hasta ese momento
- Si un elemento es mayor que el máximo, pasa a ser máximo
  - Es decir, se asigna su valor al máximo hasta el momento
  - La primera vez, se compara c_0 con -1
- Tras recorrer toda la lista, si el máximo nº encontrado es -1, C está vacía; en otro caso, es el máximo de la secuencia

Pseudocódigo 2

```
programa Maximo2
{ Máximo de una lista, posiblemente vacía }
definición de variables
  enteros: i, max, n;
  vector de enteros: c;
principio del algoritmo
  lee(n);
  para (i = 0; hasta n - 1; de uno en uno)
    lee(c[i]);
  max = -1;
  para (i = 0; hasta n - 1; de uno en uno)
    si (c[i] > max)
      max = c[i];
  si max = -1
    escribe("Lista vacía");
  sino
    escribe(max);
fin del algoritmo
```

¿Y para la posición del máximo?

Pseudocódigo 3

```
programa Maximo3
{ Posición del máximo de una lista }
definición de variables
  enteros: i, n, posmax;
  vector de enteros: c;
principio del algoritmo
  lee(n);
  lee(c[0]);
  posmax = 0;
  para (i = 1; hasta n - 1; de uno en uno)
  principio
    lee(c[i]);
    si (c[i] > c[posmax])
      posmax = i;
  fin
  escribe(posmax);
fin del algoritmo
```

¿Es necesario almacenar c?

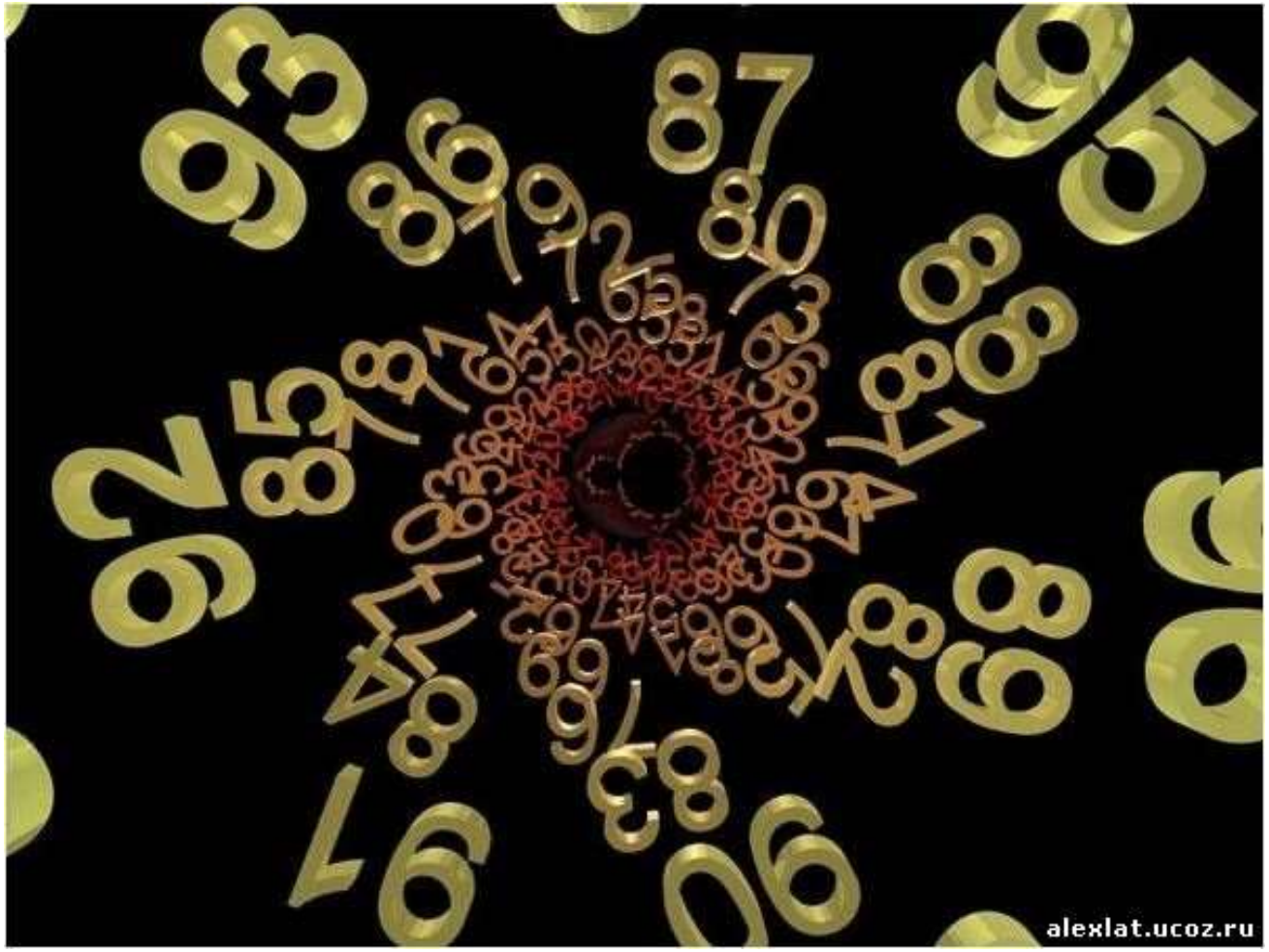
Pseudocódigo 4

```
programa Maximo4
{ Máximo de una lista, sin almacenarla en un vector }
definición de variables
  enteros: c, i, max, n;
principio del algoritmo
  max = -1;
  lee(n);
  para (i = 0; hasta n - 1; de uno en uno)
  principio
    lee(c);
    si (c > max)
      max = c;
  fin
  si max = -1
    escribe("Lista vacía");
  sino
    escribe(max);
fin del algoritmo
```

¿Y para el mínimo?

Enunciado

- Calcular el mínimo de una lista de números naturales



Pseudocódigo 5

```
programa Minimo
{ Mínimo de una lista, sin almacenarla por teclado }
definición de variables
  enteros: ci, i, min, n;
principio del algoritmo
  lee(n);
  min = ∞;
  para (i = 0; hasta n - 1; de uno en uno)
  principio
    lee(c[i]);
    si (c[i] < min)
      min = c[i];
  fin
  si min = ∞
    escribe("Lista vacía");
  sino
    escribe(min);
fin del algoritmo
```

Código en Java

¡TU TURNO!