Bubble sort

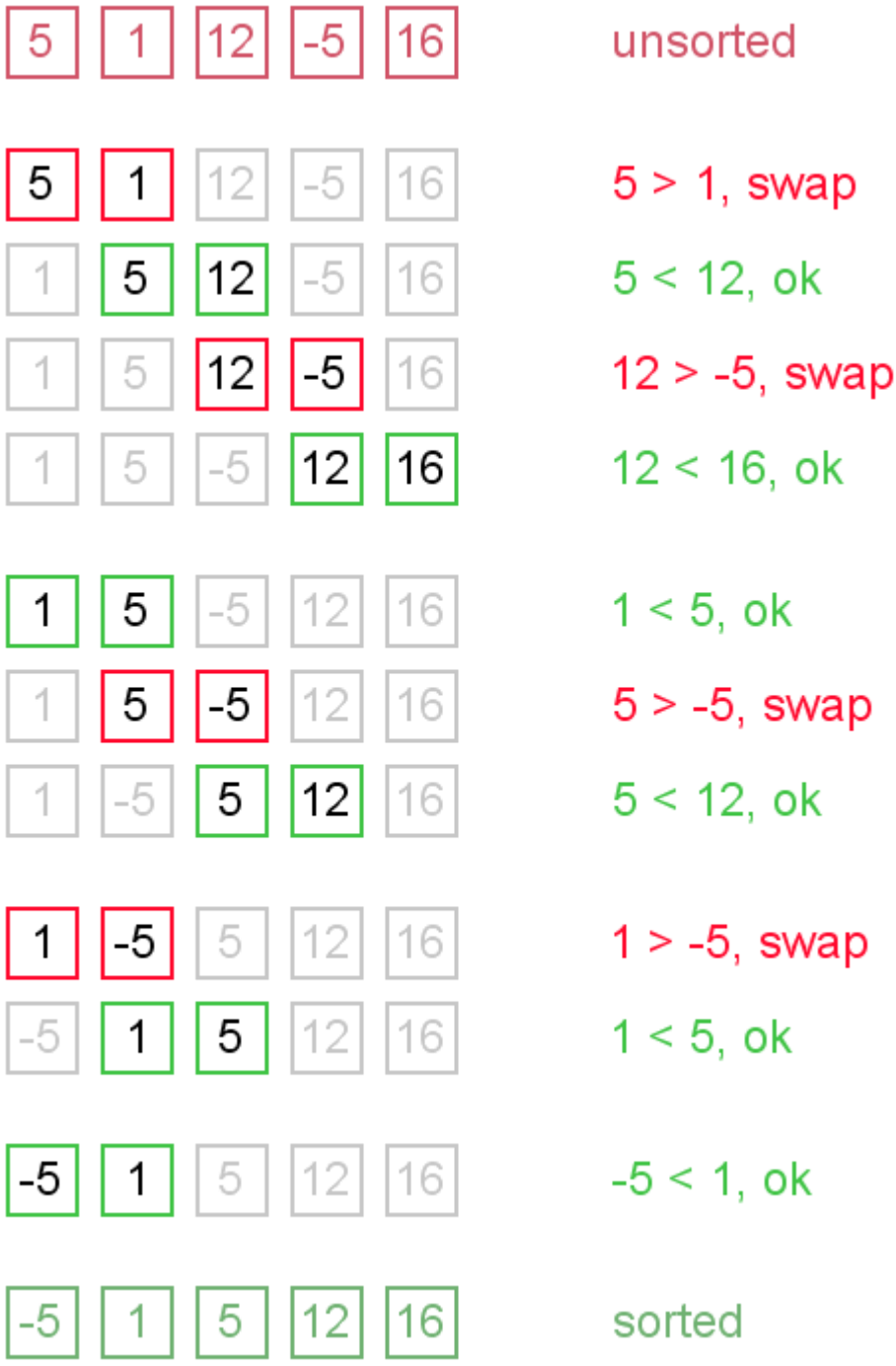
Bubble sort es un algoritmo de clasificación simple y conocido. Se usa en la práctica una vez en una luna azul y su principal aplicación es hacer una introducción a los algoritmos de clasificación. La clasificación de burbuja pertenece a los algoritmos de clasificación $O(n^2)$, lo que la hace bastante ineficaz para ordenar grandes volúmenes de datos. El tipo de burbuja es **estable** y **adaptativo** .

Algoritmo

1. Compara cada par de elementos adyacentes desde el comienzo de una matriz y, si están en orden inverso, cámbialos.
2. Si se ha realizado al menos un intercambio, repita el paso 1.

Puedes imaginar que en cada paso las burbujas grandes flotan hacia la superficie y permanecen allí. En el paso, cuando no se mueve ninguna burbuja, la clasificación se detiene. Veamos un ejemplo de ordenar una matriz para aclarar la idea de burbuja.

Ejemplo. Ordena {5, 1, 12, -5, 16} usando sort de burbuja.



Análisis de complejidad

La complejidad media y peor del caso de sortear burbujas es O (n^2). Además, hace O (n^2) swaps en el peor de los casos. El tipo de burbuja es adaptativo. Significa que para una matriz casi ordenada da una estimación de O (n). Evite las implementaciones, que no comprueban si la matriz ya está

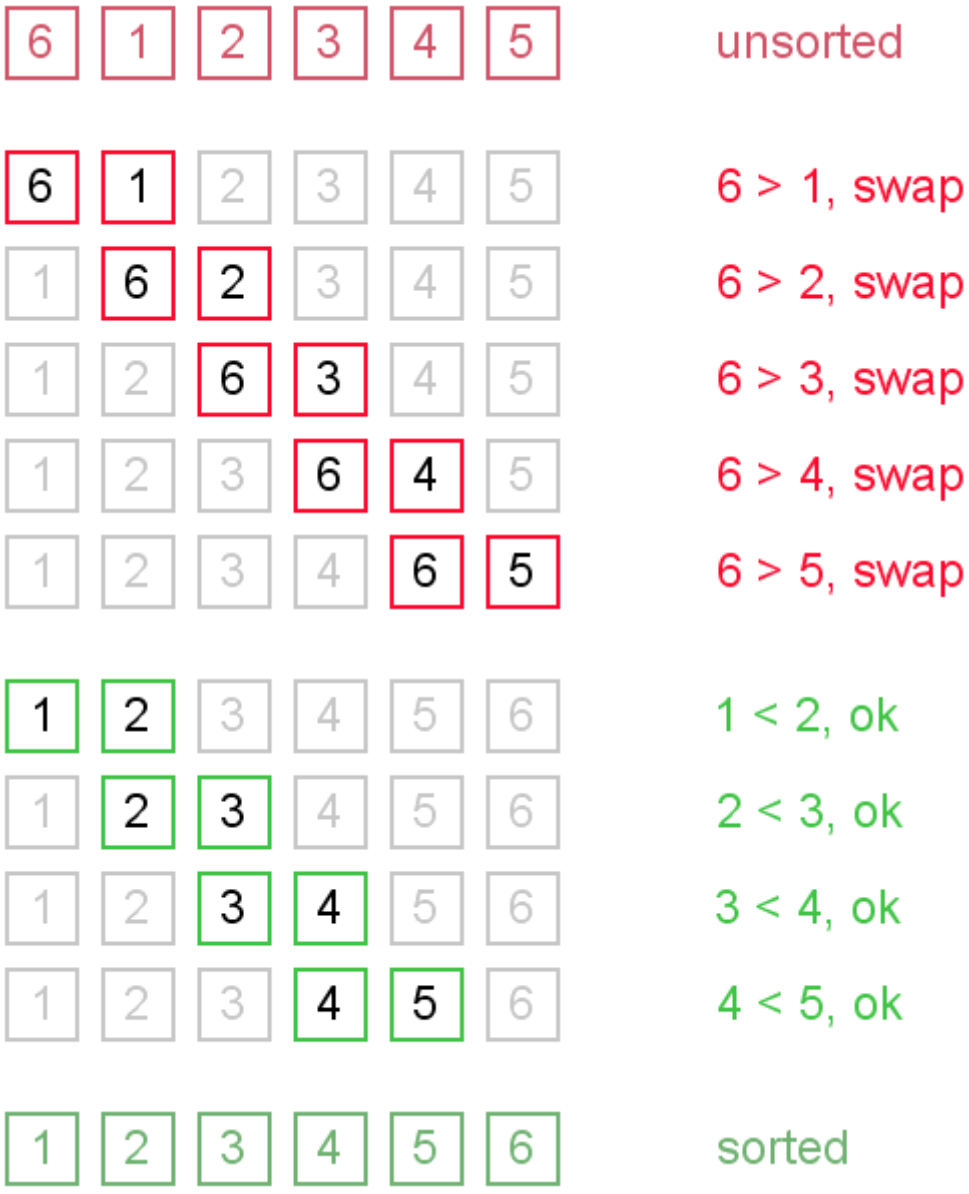
ordenada en cada paso (cualquier intercambio hecho). Esta comprobación es necesaria para preservar la propiedad de adaptación.

Tortugas y conejos

Un problema más de la clasificación de burbujas es que su tiempo de ejecución depende mucho del orden inicial de los elementos. Los elementos grandes (conejos) suben rápido, mientras que los pequeños (las tortugas) descienden muy lentamente. Este problema se resuelve en el tipo Cocktail .

Ejemplo de tortuga Pensamiento, la matriz {2, 3, 4, 5, 1} está casi ordenada, se necesitan iteraciones O (n^2) para ordenar una matriz. Elemento {1} es una tortuga.

Ejemplo de conejo La matriz {6, 1, 2, 3, 4, 5} está casi ordenada también, pero se necesitan O (n) iteraciones para ordenarla. Element {6} es un conejo. Este ejemplo demuestra la propiedad de adaptación del tipo de burbuja.



Fragmentos de código

Hay varias formas de implementar el tipo de burbuja. Tenga en cuenta que ese control de "canjes" es absolutamente necesario para preservar la propiedad de adaptación.

Java

```
public void bubbleSort ( int [] arr) {
      boolean swapped = true ;
      int j = 0;
      int tmp;
      while (intercambiado) {
```

```
            intercambiado = falso ;
            j ++;
            para ( int i   =   0;   i   <arr.length   -   j;   i   ++)
{
                  if (arr [i]> arr [i + 1]) {
                        tmp = arr [i];
                        arr [i] = arr [i + 1];
                        arr [i + 1] = tmp;
                        intercambiado = verdadero ;
                  }
            }
      }
}
```

C ++

```
void bubbleSort ( int arr [], int n ) {
      bool intercambiado = verdadero ;
      int j = 0;
      int tmp ;
      while ( intercambiado ) {
            intercambiado = falso ;
            j ++;
            para ( int i = 0; i < n - j ; i ++) {
                  if ( arr [ i ]> arr [ i + 1]) {
                        tmp = arr [ i ];
                        arr [ i ] = arr [ i + 1];
                        arr [ i + 1] = tmp ;
                        intercambiado = verdadero ;
                  }
            }
      }
}
```