Invariantes de algoritmos de sorting

Algoritmos y Estructuras de Datos II

Segundo cuatrimestre - 2010

1. Selection sort

1.1. Idea

Seleccionar el mínimo elemento, llamémoslo m, del contenedor y ponerlo al principio. Después seleccionar el mínimo elemento sin tener en cuenta m y ponerlo segundo... etc Se puede hacer in-place.

1.2. Invariante

- En la k-ésima iteración, los primeros k elementos ya están ordenados en su posición final.
- El arreglo es una permutación del arreglo original.

2. Insertion sort

2.1. Idea

Insertar el i-ésimo elemento del contenedor en la posición que le corresponde en el arreglo 0..i Se puede hacer in-place.

2.2. Invariante

- Los elementos del arreglo de 0..i son los mismos que en el arreglo original, pero están ordenados.
- El arreglo es una permutación del arreglo original.

3. Heap sort

3.1. Idea

Transformar el arreglo en un heap usando el algoritmo de Floyd en O(n). Una vez construido el heap, el máximo, llamémoslo M, se obtiene en O(1), y remover el máximo es O(log(n)). Al remover el máximo del heap, queda un espacio vació en el arreglo. Llenar ese espacio vacío, que está al final de l arreglo con M. O sea, hacélo in-place.

3.2. Invariante

- En la i-ésima iteración, los primeros n-i elementos del arreglo conforman un heap y los últimos i elementos están ordenados
- El arreglo es una permutación del arreglo original

4. Merge sort

4.1. Idea

Para ordenar un arreglo, lo parto en dos mitades A_1 y A_2. Ordeno las dos mitades A_1 y A_2 y despues las junto en O(n). Para ordenar a A_1 y a A_2 me llamo recursivamente.

No sale in-place, salvo que lo hagas sobre listas.

5. Quick sort

5.1. Idea

Elijo un elemento del arreglo, al que voy a llamar pivote. Pongo a todos los elementos menores al pivote a la izquierda y a los elementos mayores al pivote a la derecha. Me llamo recursivamente con izquierda y con derecha.

Tips: Fijate que podes elegir el pivote de muchas formas: el primer elemento, la mediana, random.