1-D

1. Aplicando merge-sort a una lista de 5 elementos, la lista se parte en dos listas de tamaño 2 y 3. Ordenar la lista de 2 resulta en una comparación. La de 3 se parte en listas de un elemento y dos elementos. Ordenando la lista de dos elementos y luego fusionar con la lista de un elemento resulta en tres comparaciones. Fusionando las dos listas ordenadas se necesita un máximo de 4 comparaciones. En total son 8 comparaciones.

Podemos reducir las comparaciones totales para encontrar la mediana cambiando la estrategia en el fusionamiento de las sublistas de 2 y 3 elementos. Para ello comparamos el primer elemento de la lista de 2 con el elemento medio de la lista de 3. Pueden pasar varias cosas:

s1 = [a,b], s2 = [c,d,e]

Compara a y d

Opción 1: a > d:

List = [c,d,a,e] y mediana = a, porque sabemos que b > a. Metiendo b no cambia la posición de a.

Opción 2: a < d:

List = [c,a,d,e],

Compara b y d:

Opción 1: b > d:

List = [c,a,d,…] y mediana = d porque comparando b y e no cambia posición de d.

Opción 2: b < d:

List = [c,a,b,d,e] y mediana = b.

Como se puede ver solo necesitamos un máximo de dos comparaciones para averiguar la media, lo que baja el número de comparaciones total a 6 como máximo.

Ein Bild, das Text, Reihe, Diagramm, Zahl enthält.

Automatisch generierte Beschreibung2. Creamos un grafo que mide el tiempo de ejecución con timeit usando listas de 1000 a 100 000 elementos ordenados de modo inverso. El crecimiento es lineal