

Quicksort szemléltetés:

Bemeneti tömb:

Array = [11, 7, 8, 4, 1, 5]

Pivot elem: utolsó

Első quicksort hívás - bemenet: Array, low = 0, high = array.length - 1

Mivel low kisebb mint high ezért megkezdjük a particionálást.

Pivot elem = Array[high] - > 5

Index (kisebb elem indexe) = low - 1 - > -1

Ciklus low-tól high-1 ig.

Mivel Array[ciklusindex] (11) nem kisebb vagy egyenlő a pivot elemnél ezért ciklus vége

Ugyanez az eset 7nél 8nál is, de a 4 az már kisebb mint 5, ezért index++ és swap(Array[index], Array[ciklusindex])

Array = [4, 7, 8, 11, 1, 5]

Következő elem az 1 es ami szintén kisebb 5 nél, az index pedig 1 lesz, ezért a 7 es kicserélődik az 1-sel és a ciklus végetér.

Array = [4, 1, 8, 11, 7, 5]

Ekkor tudjuk, hogy minden elem az Array[0:index]- ig kisebb mint a pivot elem, de az utána következő nem, ezért a pivot elemet kicseréljük ezzel. (Array[index+1](2), Array[high])

Array = [4, 1, 5, 11, 7, 8]

Return value a pivot elem végleges helyének indexe ami egyenlő lesz az index+1-el.

Ezután a particionálás előtt és után rendezzük az elemeket.

Quicksorton belüli rekurzív hívás - bemenet: Array, low = 0, high = a korábbi return value -1( -1 mert ugye az az elem már a helyén van)

Mivel low=0 kisebb mint high=1 ezért elkezdünk particionálni a tömb bal részében.

Pivot elem : 1 es (Array[high])

Ciklus 0 tól 1 ig, de mivel a 4 es elem nem kisebb az 1 esnél nem történik semmi a ciklusban.

Partición belüli indexünk így tehát maradt -1.

A 4 es-es kicserélődik az 1 esszel mert swap(Array[index+1], Array[high])

Ezzel vége is a bal oldali particionálásnak, mert nincs mit tovább particionálni, a további rekurzív hívásoknál nem igaz a low < high, mert jelen esetben 0 -1, majd 1 1 lenne

Array = [1, 4, 5, 11, 7, 8]

Második rekurzív hívás, most a tömb 'jobb' oldalában particionálunk.

Low = első particionálás return értéke + 1(mert ugye a returned value már rendezett elem(5ös)->3

High = Array.length -1 ->5

low < high ezért megkezdjük a particionálást

Pivot elemünk: 8-as (Array[high])

Ciklus 3 tól 5 ig...

A 11 es ugye nem kisebb 8 nál, de a 7 es igen ezért a particion belüli indexünk ami 2 ről indult 3 lesz és történik egy csere a 3. és 4. elem között, majd a ciklus végetér

Array = [1, 4, 5, 7, 11, 8]

Ezután ugye még a pivot elem kicserélődik az index+ 1 elemmel, tehát swap(Array[index+1])(4), pivotelem)

Array = [1, 4, 5, 7, 8, 11]

Ezzel véget is ér a második rekurzív hívás és ezzel a quicksort function is, mert a jobb oldalon a 2 rekurzív hívásnál nem igaz a low < high, mert jelen esetben a 'bal' oldali rendezésnél a low =3, high = 3, 'jobb' oldalon pedig low =5, high = 5.

Végeredmény tehát:

[1, 4, 5, 7, 8, 11]