Problema 1

Algorithm 1 HeapSort

```
1: procedure HEAPSORT(A)
2:     BUILD-MAX-HEAP()
3:     for i ← 1 to len(A) − 1 do
4:         for j ← 2 to len(A) − 1 do
5:             if A[i] <= A[j] then
6:                 min ← A[j]
7:         newArr [i] = min
```

running-time: es peor que el HeapSort porque su complejidad es de O(n^2)

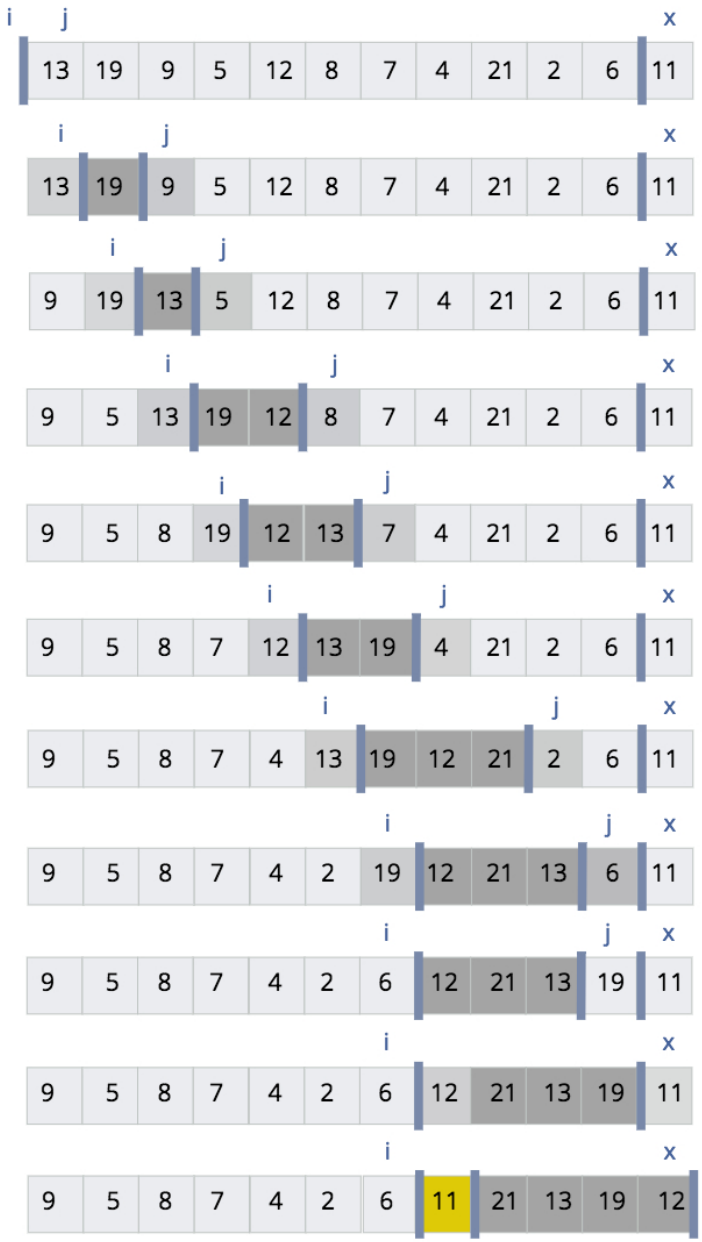
Problema 2

1.) O($n * log(n)$)

3.) El HeapSort siempre será complejidad de n log(n), sin embargo el Quicksort tiene la ventaja de hacer swaps necesarios, no como el HeapSort que siempre rompe la propiedad de heap y la vuelve a construir. En el Quicksort también se tiene la ventaja de elegir un buen pivot y posiblemente ahorre trabajo de partición o casualmente el pivote elegido es un valor intermedio de los valores.

Problema 2

2.)


i j x
13 19 9 5 12 8 7 4 21 2 6 11
i j x
13 19 9 5 12 8 7 4 21 2 6 11
i j x
9 19 13 5 12 8 7 4 21 2 6 11
i j x
9 5 13 19 12 8 7 4 21 2 6 11
i j x
9 5 8 19 12 13 7 4 21 2 6 11
i j x
9 5 8 7 12 13 19 4 21 2 6 11
i j x
9 5 8 7 4 13 19 12 21 2 6 11
i j x
9 5 8 7 4 2 19 12 21 13 6 11
i j x
9 5 8 7 4 2 6 12 21 13 19 11
i x
9 5 8 7 4 2 6 12 21 13 19 11
i x
9 5 8 7 4 2 6 11 21 13 19 12

Problema 3

```python
def QuickSort(array,p,r):
    if p < r:
        q = Partition(array, p,r)
        QuickSort(array, p, q-1)
        QuickSort(array,q+1,r)

def Partition(array, p, r):
    i = (p - 1)  # indice del elemento mas pequeño
    pivot = array[r]  # pivot

    for j in range(p, r):

        #Si el elemento actual es menor o igual al pivot
        if array[j] <= pivot:
            #Incrementa el indice del elemento menor
            i = i + 1
            array[i], array[j] = array[j], array[i]

    array[i + 1], array[r] = array[r], array[i + 1]
    return (i + 1)




unsorted_array = [5, 10, 15, 32, 55, 21, 40, 2, 3, 76, 89, 28, 9, 7]

QuickSort(unsorted_array, 0, len(unsorted_array) - 1)
print unsorted_array
```

QuickSort() › if p < r

```
ickS
 /Library/Frameworks/Python.framework/Versions/2.7/bin/python2.7 /Us
 [2, 3, 5, 7, 9, 10, 15, 21, 28, 32, 40, 55, 76, 89]

 Process finished with exit code 0
```