

ELİF ÇELİK 180401055

Ders videoları kodlarının GitHub linki :

<https://github.com/eelifcelik/Programlama-Lab/tree/master/Uzaktan%20Egitim/Hafta4>

****min_heapify:** Bir array ve i indisi alarak bu arraydaki i indisine kadar olanları heap yapısı olarak düzenliyor.

****build_min_heap:** Bir array alıyor ve arrayin bazı elemanlarını(roottan başlayıp n/2 ye kadar olanlarını) min_heapify fonksiyonuna göndererek heapi oluşturuyor.

****heapsort:** Parametre olarak bir array alır, kopyasını oluşturur ve bu ağacın en üstündeki sayıyı alarak sıralama işlemi yapar.

****insertItemToHeap:** Bir Minheap dizisine eleman eklemeyi sağlıyor.

****removeItemFrom:** Bir MinHeap dizisinden minHeap oluşunu bozmadan eleman silmeyi sağlıyor.

```
def min_heapify(array, i):
    left = 2 * i + 1
    right = 2 * i + 2
    length = len(array) - 1
    smallest = i
    if left <= length and array[i] > array[left]:
        smallest = left
    if right <= length and array[smallest] > array[right]:
        smallest = right
    if smallest != i:
        array[i], array[smallest] = array[smallest], array[i]
        min_heapify(array, smallest)
```

```
def build_min_heap(array):
    for i in reversed(range(len(array)//2)):
        min_heapify(array, i)
```

```
def heapsort(array):
    array = array.copy()
    build_min_heap(array)
    sorted_array = []
    for _ in range(len(array)):
        array[0], array[-1] = array[-1], array[0]
        sorted_array.append(array.pop())
        min_heapify(array, 0)
    return sorted_array
```

```
def insertItemToHeap(myheap_1,item):  
    n=len(myheap_1)  
    myheap_1.append(item)#once elemanı ekliyoruz
```

```
    if(myheap_1[n]>=myheap_1[n//2]):  
        print(myheap_1)  
        return myheap_1
```

```
    else:  
        while(myheap_1[n]< myheap_1[n//2]):  
            gecici = myheap_1[n]  
            myheap_1[n]=myheap_1[n//2]  
            myheap_1[n//2]= gecici  
            print(myheap_1)
```

```
    return myheap_1
```

```
#liste sonunundan eleman silmek için  
#def removeItemFrom(myheap_1):  
#    n=len(myheap_1)  
#    myheap_1.pop(myheap_1[n-1])  
#    print(myheap_1)
```

```
def removeItemFrom(myheap_1):#Heapten eleman siler.  
    length = len(myheap_1)  
    if length == 0:  
        print("Heap boş ...")  
        return myheap_1  
    heapArray = heapsort(myheap_1)  
    heapArray[0],heapArray[-1] = heapArray[-1],heapArray[0]  
    heapArray.pop()  
    build_min_heap(heapArray)  
    return heapArray
```

```
my_array_1=[8,10,3,4,7,15,1,2,16]  
build_min_heap(my_array_1)  
print(my_array_1)
```

```
my_array_2=heapsort(my_array_1)  
print(my_array_1, my_array_2)
```

```
removeItemFrom(my_array_2)
```