

Tri par tas

vendredi 14 avril 2017

14:35

1)

```
def heapsort(v):  
    tas = []  
    while v != []:  
        heappush(tas, v.pop())  
    while tas != []:  
        v.append(heapop(tas))
```

2) Coût : $O(n \log n)$

3) Méthode diviser pour régner

```
def reorganiser(v, racine):  
    if racine*2 + 1 >= len(v):  
        return  
    reorganiser(v, 2*racine + 1)  
    reorganiser(v, 2*racine + 2)  
    stalagtite(v, racine)
```

$$4) C(n) = 2C\left(\frac{n}{2}\right) + O(\log n)$$

D'après le master theorem, c'est un coût en $O(n^{\log_2 2}) = O(n)$