

Binary Heap Homework 2

- Consider the next algorithm:

```
def Ex2 ( A ):  
    D ← build ( A )  
    while ¬ is_empty ( D )  
        extract_min ( D )  
    endwhile  
enddef
```

where A is an array.

Compute the time-complexity of the algorithm when:

- build, is_empty ∈ $\Theta(1)$, extract_min ∈ $\Theta(|D|)$;
- build ∈ $\Theta(|A|)$, is_empty ∈ $\Theta(1)$, extract_min ∈ $O(\log n)$;

Solution

- ```
def Ex2 (A):
 D ← build (A) # $\Theta(1)$
 while ¬ is_empty (D) # at each iteration $\Theta(1)$
 extract_min (D) # $\Theta(|D|)$
 endwhile # at each iteration one element is removed -> $|D|$ iterations
enddef
```

$$\begin{aligned} T_1(|A|) &= \Theta(1) + \sum_{i=0}^{|D|-1} (\Theta(1) + \Theta(|D| - i)) = \Theta(1) + \Theta\left(\sum_{i=0}^{|D|-1} |D| - i\right) \\ &= \Theta(1) + \Theta\left(\sum_{i=1}^{|D|} i\right) = \Theta(1) + \Theta\left(\frac{|D|(|D| - 1)}{2}\right) \\ &= \Theta(|D|^2) \end{aligned}$$

- ```
def Ex2 ( A ):  
    D ← build ( A ) #  $\Theta(|A|)$   
    while ¬ is_empty ( D ) #  $\Theta(1)$   
        extract_min ( D ) #  $O(\log n)$   
    endwhile # at each iteration one element is removed ->  $|D|$  iterations  
enddef
```

$$\begin{aligned} T_2(|A|) &= \Theta(|A|) + \sum_{i=0}^{|D|-1} (\Theta(1) + O(\log(|D| - i))) \\ &= \Theta(|A|) + O\left(\sum_{i=1}^{|D|} \log(|D|)\right) = \Theta(|A|) + O\left(\sum_{i=1}^{|D|} \log i\right) \\ &= \Theta(|A|) + O\left(\log\left(\prod_{i=1}^{|D|} i\right)\right) = \Theta(|A|) + O(\log(|D|!)) \end{aligned}$$