# Binary Heaps: homework 2

Eros Fabrici

Course of AA 2019-2020

## Ex. 2

- Let $n$ be the length of $A$ (and thus of $D$). —extract_min— will reduce the length of $D$ by one, therefore the while loop will iterate $n$ times. Thus we can reduce our complexity function as

  $$\sum_{i=1}^{n} \theta(i) = \theta(\sum_{i=1}^{n} i) = \theta(\frac{n}{2}(n+1)) = \theta(n^2)$$

  Thus $T(n) \in \theta(n^2)$

- Let $n$ be the length of $A$ (and thus of $D$). As before, the while loop will iterate $n$ times and thus we have

  $$\theta(n) + \mathcal{O}(\sum_{i=1}^{n} log(i))$$

  Now let's focus on $\sum_{i=1}^{n} log(i)$. We can observe that

  $$\sum_{i=1}^{n} log(i) < \sum_{i=1}^{n} log(n) = n\ log(n)$$

  A lower bound may be defined by Riemann sums

  $$\sum_{i=1}^{n} log(i) > \int_{1}^{n} log(x)dx = n\ log(n) - n + 1$$

  Therefore $\sum_{i=1}^{n} log(i) \in \theta(n\ log(n))$.
  Back to our problem, we have then

  $$\theta(n) + \mathcal{O}(n\ log(n)) = \mathcal{O}(n\ log(n))$$

  Therefore $T(n) \in \mathcal{O}(n\ log(n))$.