# ALGORITHMIC DESIGN - HOMEWORK 3 - BINARY HEAPS 2

KEERTHANA C J

September 2020

## 1 Heap without swapping keys

The binary heap has been implemented without the necessity of swapping the keys with the help of two arrays. The time needed to delete the minimum from the heap is compared for the two implementations. The time needed is

| Size of instance | Heaps | Heaps without swapping keys |
|---|---|---|
| 0 | 0.000009 | 0.000027 |
| 1820 | 0.079779 | 0.674651 |
| 3640 | 0.120430 | 2.158138 |
| 5461 | 0.216084 | 4.485764 |
| 7281 | 0.273035 | 8.136988 |
| 9102 | 0.346384 | 12.792705 |
| 10922 | 0.391069 | 18.218128 |
| 12743 | 0.565134 | 28.181985 |
| 14563 | 0.600841 | 39.668851 |
| 16384 | 0.700654 | 52.495281 |

Table 1: Time needed to extract the minimum value from a heap vs from a heap without swapping keys

considerably higher and is nearer to that of extracting the minimum value from an array.

## 2 Complexity

1. When is_empty $\in \Theta(1)$, extract_min $\in \Theta(|D|)$
   **Answer**: Complexity $= \Theta(1) + |D|.\Theta(|D|) = \Theta(|D|^2) = \Theta(|A|^2) \simeq$ Selection Sort
2. When build $\in \Theta(|A|)$ is_empty $\in \Theta(1)$, extract_min $\in O(log|D|)$
   **Answer**: Complexity $= \Theta(|A|) + \Theta(1) + |D|.O(log|D|) = O(|A|log|A|) \simeq$ Heap Sort