Homework 3: Binary Heaps(2)

Azza M. N. Abdalghani

Exercise 1

By modifying the code written during the last lessons, provide an array-based implementation of binary heaps which avoids to swap the elements in the array A.
(*Hint*: use two arrays, key pos and rev pos, of natural numbers reporting the position of the key of a node and the node corresponding to a given position, respectively)

The new implementation can be found in `AD_bin_heaps` folder, the new arrays `unsigned int *key_pos` and `unsigned int *rev_pos` are added to the struct of binary heaps. Accordingly, `swap_keys` function has been modified and all other related functions.

Exercise 2

Consider the next algorithm:

```
def Ex2(A)
  D ← build(A)
  while ¬ is_empty(D)
     extract_min(D)
  end while
```

where `A` is an array. Compute the time-complexity of the algorithm when:

- `build`, `is_empty` $\in \Theta(1)$, `extract_min` $\in \Theta(|D|)$;

  In this case, first statement `build(A)` costs $\Theta(1)$ and the while condition costs the same too, `extract_min(D)` costs $\Theta(|D|)$ but it will be executed $|D|$ times. Then the total cost will be :

$$T(|D|) = \Theta(1) + \sum_{i=1}^{|D|} (\Theta(1) + \Theta(|D|)) = \Theta(1) + \Theta(|D|) + \Theta(|D|^2) \in \Theta(|D|^2)$$

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

Here `build(A)` costs $\Theta(|A|)$ and as before while condition costs $\Theta(1)$, and there is $|D|$ loops for `extract_min(D)` statement which costs $\in O(\log |D|)$. Thus, observing that $|A| = |D|$ and by assuming that $|A| = |D| = n$, then the total cost is :

$$T(n) = \Theta(n) + \sum_{i=1}^{n}(\Theta(1) + O(\log n)) = \Theta(n) + O(n \log n) \in O(n \log n)$$