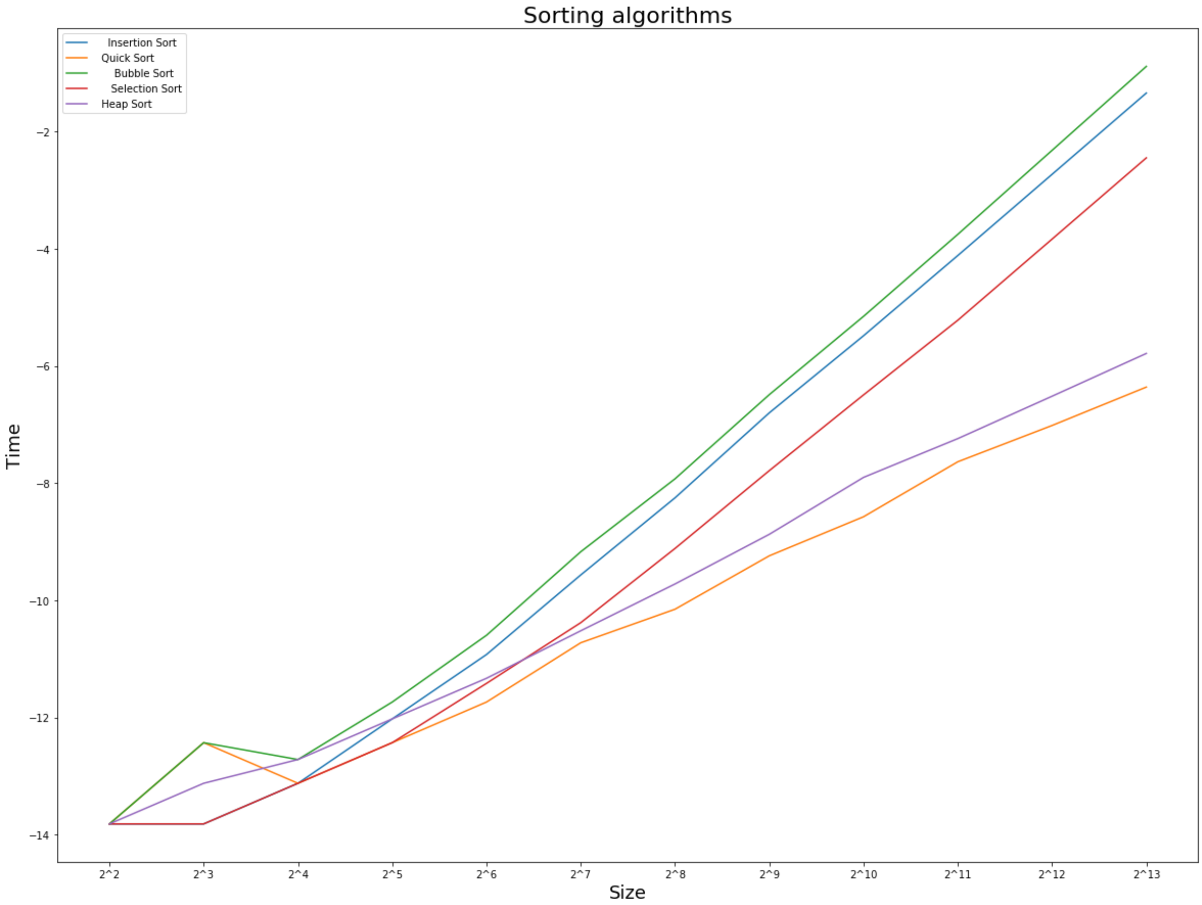
Sorting Homework

Exercise 2

In the plot we can see the logarithm transformation of the execution time in relation with the input size of the Insertion Sort, Quick Sort, Bubble Sort, Selection Sort, and Heap Sort. As we can see The Bubble Sort has the worst performances whereas the Quick Sort is the bests.



Exercise 3

Argue about the following statement and answer the questions

1. Heap Sort on a array A whose length is n takes time O(n).
2. Heap Sort on a array A whose length is n takes time Ω(n).
3. What is the worst case complexity for Heap Sort?
4. Quick Sort on a array A whose length is n takes time O(n^3).
5. What is the complexity of Quick Sort?
6. Bubble Sort on a array A whose length is n takes time Ω(n).
7. What is the complexity of Bubble Sort?

Solution

1. FALSE. Heap Sort build an Heap (cost $\Theta(n)$) and then extract the min (cost $O(\log i)$) and repeat this until the heap is not empty ($i = n..2$) . The total complexity is

   $T_H(n) = \Theta(n) + \sum_{i=2}^{n} O(\log i) = O(n \log n)$

   that is to say that, in the worst case scenario, it cannot perform better then $n \log n$. Since $n \log n > n$ ($n > 0 \Rightarrow \log n > 1$), Heap Sort cannot take time $O(n)$.

2. TRUE: In the case of an already sorted array (best case scenario), the extraction of the min takes time $\Theta(1)$ (no need to call Heapify).

   $T_H(n) = \Theta(n) + \sum_{i=2}^{n} \Theta(1) = \Theta(n)$

   Hence Heap Sort can take time $\Omega(n)$.
3. The worst case complexity for Heap Sort is $\Theta(n \log n)$.
4. TRUE: Given that the complexity of the Quick Sort is $O(n^2)$, it is also true that it is $\in O(n^3)$.
5. The complexity of the Quick Sort depends on the choice of the pivot. Nevertheless we know that $\Theta(n \log n) \leq T_{QS} \leq \Theta(n^2)$.
6. TRUE: Given that the complexity of the Bubble Sort is $\Theta(n^2)$, it is also true that $T_{BS} \in \Omega(n^2)$. Given the definition of $\Omega(\cdot)$ it is also true that $T_{BS} \in \Omega(n)$.
7. The complexity of the Bubble Sort is $\Theta(n^2)$.

Exercise 4

Solve the following recursive equation:

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 32 \\ 3 * T(\frac{n}{4}) + \Theta(n^{3/2}) & \text{otherwise} \end{cases}$$

Solution

There are 3 recursive call at first step, 9 at the second step and so on. At the *i-th* step there will be 3^i recursive calls and each one will take time $c(\frac{n}{4^i})^{\frac{3}{2}}$. Given the base case $\Theta(1)$ when $n = 32$, the height of the recursion tree will be $\frac{1}{2}\log_2(\frac{n}{32}) = \frac{1}{2}(\log_2 n - log_2 32) = \frac{1}{2}(\log_2 n - 5)$.

$$\begin{aligned} T(n) = & \sum_{i=0}^{\frac{1}{2}(\log_2 n-5)-1} 3^i c(\frac{n}{4^i})^{\frac{3}{2}} + \Theta(n^{\frac{1}{2}\log_2 3}) = \sum_{i=0}^{\frac{1}{2}(\log_2 n-5)-1} 3^i c\frac{n^{\frac{3}{2}}}{4^{(i\frac{3}{2})}} + \Theta(n^{\frac{1}{2}\log_2 3}) = cn^{\frac{3}{2}} \sum_{i=0}^{\frac{1}{2}(\log_2 n-5)-1} (\frac{3}{2^3})^i + \Theta(n^{\frac{1}{2}\log_2 3}) \\ & \leq cn^{\frac{3}{2}} \sum_{i=0}^{+\infty} (\frac{3}{2^3})^i + \Theta(n^{\frac{1}{2}\log_2 3}) = cn^{\frac{3}{2}} \frac{1}{1-\frac{3}{2^3}} + \Theta(n^{\frac{1}{2}\log_2 3}) \\ & \leq \frac{8}{5}cn^{\frac{3}{2}} + \Theta(n^{\frac{1}{2}\log_2 3}) \in O(n^{\frac{3}{2}}) \end{aligned}$$