Assignment 3

Jixuan Ruan PB20000188

September 29, 2022

1. stable: divide and conquer, counting sort
   unstable: insertion sort, heap sort, quick sort
   my solution: Add a present order number to every element taking a role in the sorting process. If the value of two elements is proved to be equal, then first check the order number and later decide whether to exchange the two elements. The extra space complexity is $\Theta(nlgn)$, and the time complexty remains the same.

2. Firstly, we can assume that the digit length of the integer can be $b_1.....b_l$ and each length matches num_i integer.

$$\sum_{i=1}^{l} num_i b_i = n$$

   My solution:
   1) divide the array into l groups with the same digit length.
   2) use radix sort in each group.
   3) merge all the group in order into a complete one.
   Step1 costs $\Theta(n)$ time complexity. Step2 costs $\sum_{i=1}^{l} \Theta(b_i num_i)$ time complexity, which equals to $\Theta(n)$.
   So the time complexity of this algorithm is $\Theta(n)$

3. My solution(while $i < \frac{n}{2}$, else use randomized-select):
   1) divide the array into $\lceil \frac{n}{2} \rceil$ groups.
   2) find the smaller one of each group.
   3) find the first i elements in the smaller array.
   4) use the randomized-select to find the ith element from groups containing the elements found in step3.

$$T(n) = \lceil \frac{n}{2} \rceil + T(\lfloor \frac{n}{2} \rfloor) + 2i$$

   We can see that when $i < \frac{n}{2}$, we assume that the $T(n) = n + O(lgn)$, which means that $T(n) < n + clgn$:

$$\begin{aligned} T(n) &= \lceil \frac{n}{2} \rceil + T(\lfloor \frac{n}{2} \rfloor) + 2i \\ &\leq \lceil \frac{n}{2} \rceil + \lfloor \frac{n}{2} \rfloor + clg\lfloor \frac{n}{2} \rfloor + 2i \\ &= n + clg\lfloor \frac{n}{2} \rfloor + 2i \\ &\leq n + clgn \end{aligned}$$

   So we can see that we need only $n + O(lgn)$ comparison.