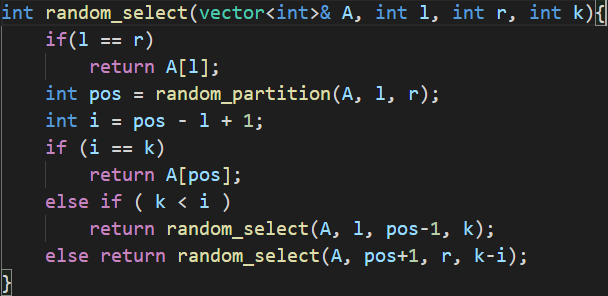
Programming Assignment #1

Find the Kth smallest element in an input array

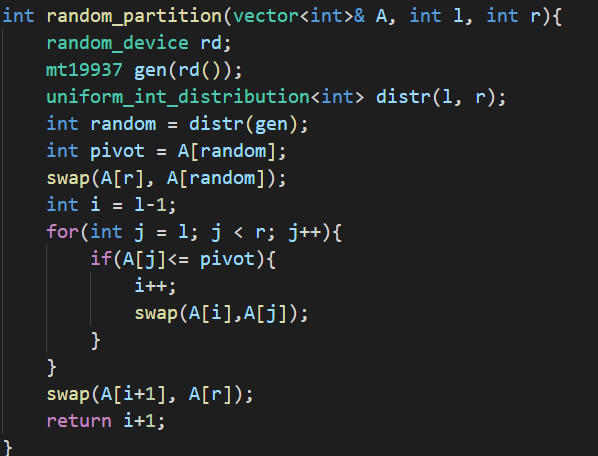
108062135 呂佳恩

Pseudo code:

Randomized-select :

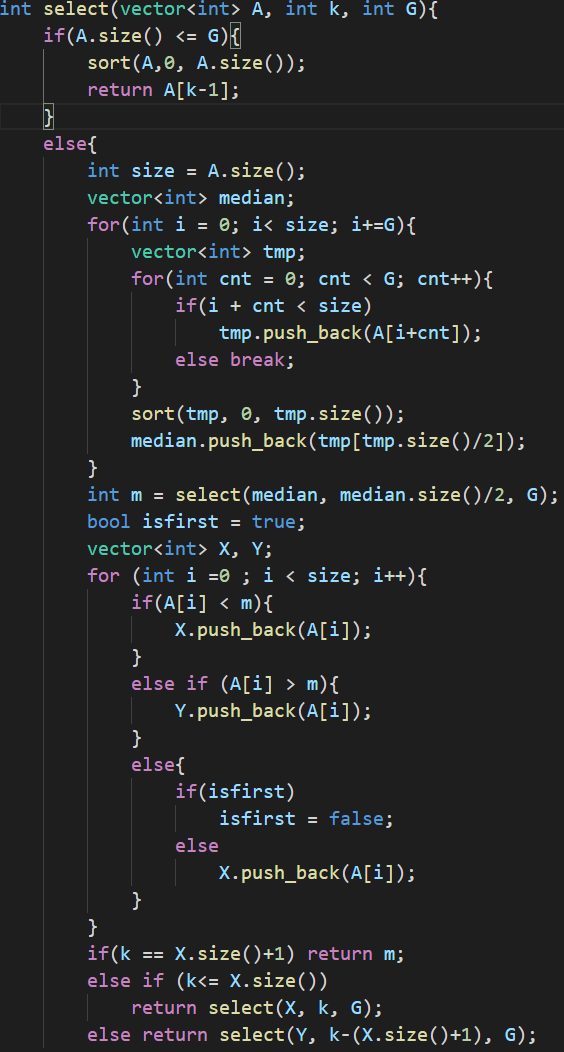


Random\_partition :



The approach for the randomized select is similar to the pseudo code provided in the slides of the class. Use the random library provided by C++. Choose a random pivot partition the array into left and right.

Median-of-Median



If the size is smaller then the partition, we just sort it and return the wanted rank.

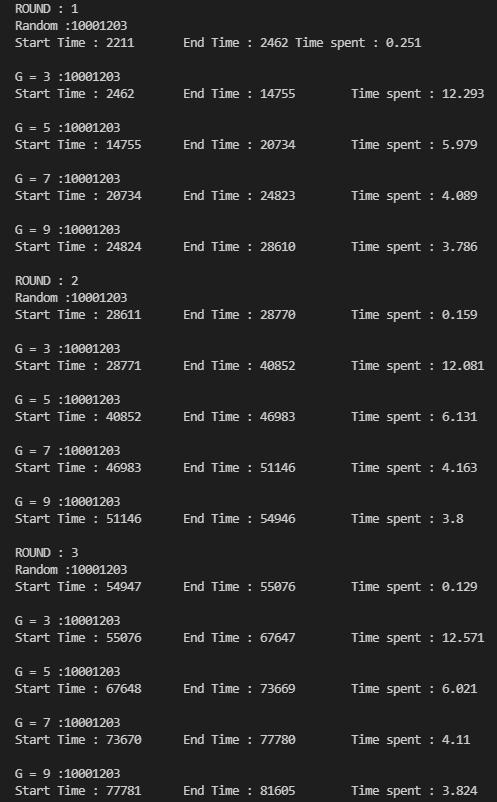
Otherwise, we have to partition the array, and then sort to get the median. We read the array, then sort every partition. After Obtaining the median, we push it in an median\_vector. We then get the median of the median\_vector by repeating the select move.

The thing we have to consider is that if there are multiple numbers with the same value. We set a flag to record if the value has existed before. Otherwise, the approach is similar to the pseudo code provided in class.

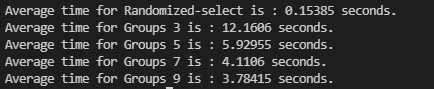
Running Time of Algorithm.

For N = 10, 000, 000, K = 5, 000, 000. We run twenty rounds. I have snipped the result of the first three rounds and the average run time. It is the images below.

The first three running rounds

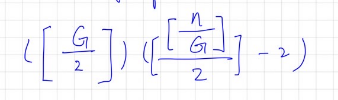


The Average running time.



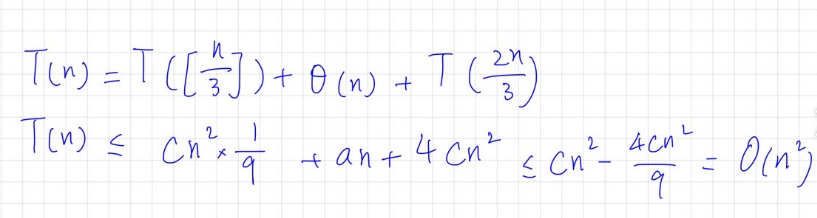
As the data above suggests, the running time of Randomized-select algorithm is much faster than the Medians-of-Medians algorithm.

Time complexity Analysis

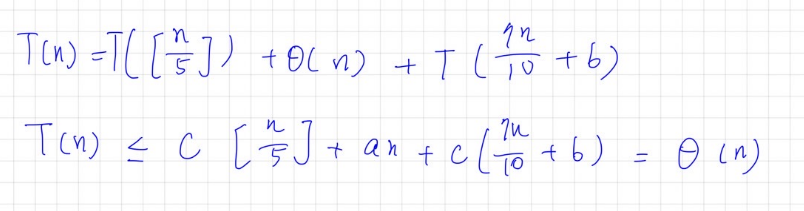
If we divide into G groups

The number of items smaller than M is at least

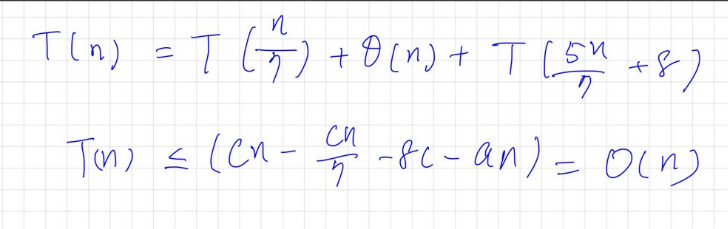
Into groups 3



Groups 5



Groups 7



Groups 9

