

Selection

Introduction

The purpose of this program is to select the element of rank k (i.e. the k 'th smallest element) from an input array. The array is written in *input.txt*, with its size in the first line and the rank k in the second line. The output would be saved in *output.txt*.

Concept

1. Read Input Array

- (1) Read the first line of *input.txt* to get the array size n .
- (2) Read n consecutive numbers from the file and store them into vector *vec*.
- (3) Note that the numbers are stored beginning from *vec*[1] instead of *vec*[0]. This is for simplification in the following process, since the index of the pivot after *Partition* will just be its rank.

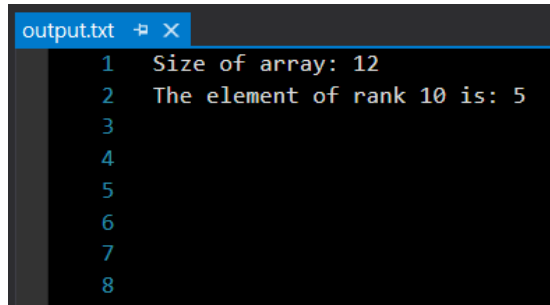
2. Do Selection

A recursive method is applied to solve the issue. The process of searching for the element of rank k goes through the following steps:

- (1) To scale down the size of the problem, use *MedianOfMedians* to divide the input array into groups of 5 elements (the last group might have fewer than 5), and then find the median of each group. This can be done by calling *Selection* to select the element of rank 3 in the group. The medians are moved to the front of the array, so as to use *Selection* recursively to find the median-of-medians.
- (2) As soon as the median-of-medians is found, choose it as the pivot and partition around it using *Partition*. Note that since *Partition* chooses the last element as pivot, the median-of-medians has to be swapped with the last element in advance.
- (3) After *Partition*, all the elements before the pivot are smaller than it, while the ones behind it are larger. This means that now the pivot's index is exactly the same as its rank. Thus, if the pivot's index equals to the rank k we are searching for, then it should be our target, so the index is returned. Otherwise, if k is smaller or larger than pivot's index, recursively call *Selection* on the smaller or larger section respectively.

3. Write Output Array

The result is written in *output.txt* in the following format:



A screenshot of a text editor window titled "output.txt". The window has a dark background with light blue text. The content of the file is as follows:

```
1 Size of array: 12
2 The element of rank 10 is: 5
3
4
5
6
7
8
```

. Divide the n elements of the input array into $b_{n=5c}$ groups of 5 elements each and at most one group made up of the remaining $n \bmod 5$ elements. 2. Find the median of each of the $d_{n=5e}$ groups by first insertion-sorting the elements of each group (of which there are at most 5) and then picking the median from the sorted list of group elements. 3. Use SELECT recursively to find the median x of the $d_{n=5e}$ medians found in step 2. (If there are an even number of medians, then by our convention, x is the lower median.) 4. Partition the input array around the median-of-medians x using the modified version of PARTITION. Let k be one more than the number of elements on the low side of the partition, so that x is the k th smallest element and there are n_k elements on the high side of the partition. 5. If $i \leq k$, then return x . Otherwise, use SELECT recursively to find the i th smallest element on the low side if $i \leq k$.