**Report for Algorithm HW1**

* **Tashi Tengyal, 2020-15441**

Environment:

Python 3.9.12 main

[Clang 13.1.6 (clang-1316.0.21.2)] on Darwin

\*Directions to run the program is in **Readme.txt**

Pseudocode for Checker Program:

bool checker(arr, result, rank) {

if(result not in arr) {

return "Result not in input array"

}

counter = 0

for(i = 0; i <= length(arr); i++ ) {

if(arr[i] <= result) {

counter += 1

}

}

if(counter == rank) {

return True

}

else {

return False

}

}

**Arguments:**

|  |  |
| --- | --- |
| Data type | Name |
| List | Arr |
| Float | Result |
| Rank | Int |

**Explanation:**

The checker program first checks if the result is in arr (input array). If it is, then it proceeds on to count the number of elements in arr that are less than or equal to result given by either random selection or deterministic selection and stores it in a counter variable. Finally, it compares the rank to the counter. If they are equal that means the result is correct and false otherwise.

**Example for Checker:**

Given the arguments below:

arr = [2, 6, 9, 1, 5]

rank = 3

result = 5

There are three elements less than or equal to 5 in arr. So, after the end of the for loop, counter = 3 which is equal to the rank we are looking for. Therefore verifies and returns that the result is indeed True.

**Time complexity for Checker Program**

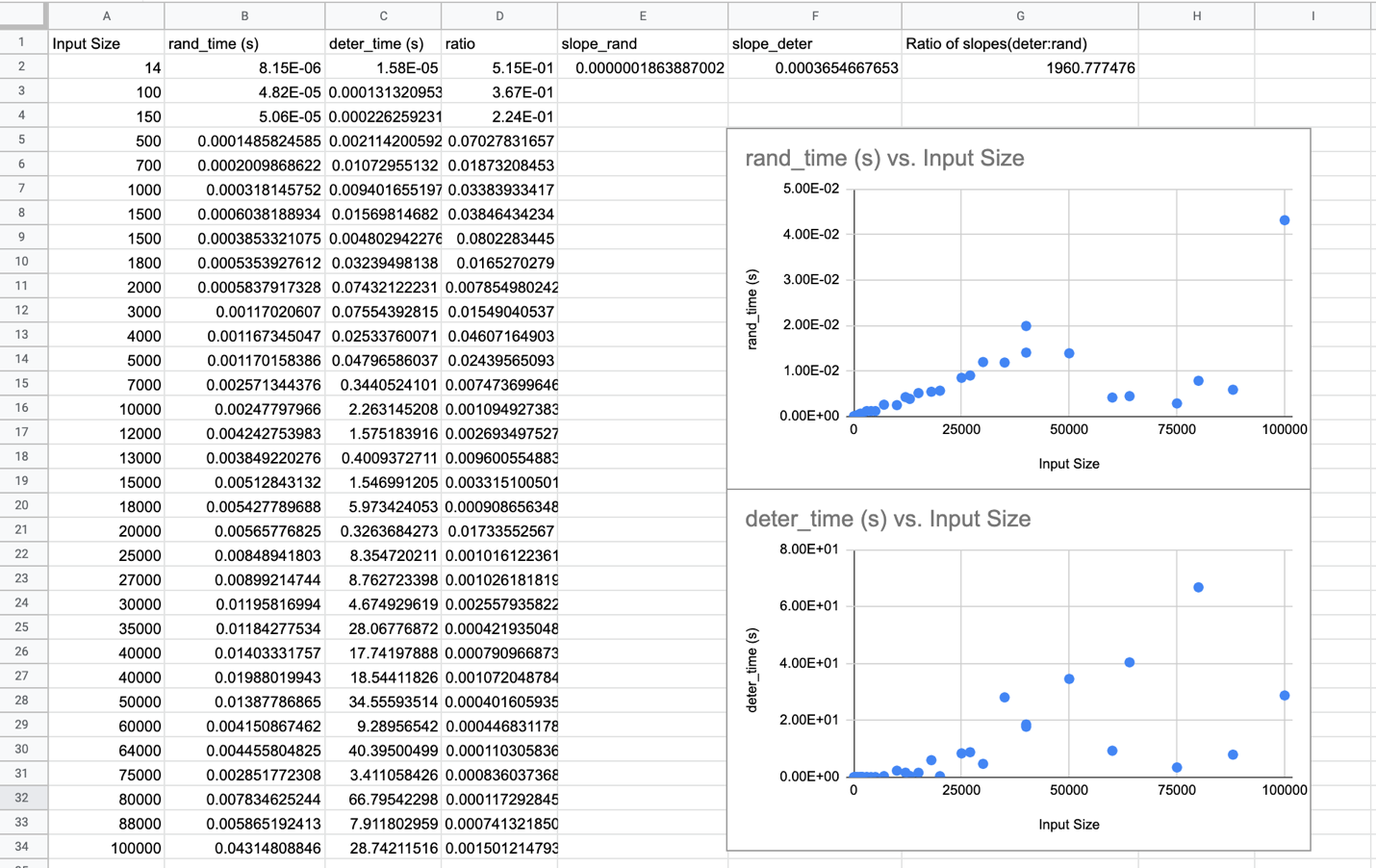
Since comparison of array elements to just one value entirely depends on the number of elements present in the array, the time complexity is linear O(n)

**Time analysis for Random Selection**

\*The excel file is attached in the directory.

A total of 33 input files of various sizes were ran using main.py. As shown below, the constants of the linear functions are reflected as slopes of the respective algorithm with the ratio between the two also displayed. The graphs obtained also reflects a somewhat linear relationship.

For the purpose of analysis, each input file was run for 10 times and taken the average of the running times. However, for more accurate results, a huge number of runs for each file would be required.



**Example Running**

1/

Input File = input1.txt

Number of elements = 14

Searching for rank = 3

Random Selection Output = 1.5

Time taken by Random Selection = 2.3126602172851562e-05 s

Deterministic Selection Output = 1.5

Time taken by Deterministic Selection = 5.412101745605469e-05 s

Checker for Random Selection = True

Checker for Deterministic Selection = True

2/

Input File = input9.txt

Number of elements = 20,000

Searching for rank = 330

Random Selection Output = 18297.0

Time taken by Random Selection = 0.004842996597290039 s

Deterministic Selection Output = 18297.0

Time taken by Deterministic Selection = 0.7167961597442627 s

Checker for Random Selection = True

Checker for Deterministic Selection = True