<u>Αναφορά Παράδοσης (εργασία 2)</u>

- Α
- Β
- Γ

Για το μέρος Γ χρησιμοποιήθηκε ο αλγόριθμος quick sort.

```
/* low  -> Starting index,  high  -> Ending index */
quickSort(arr[], low, high) {
  if (low < high) {
    /* pi is partitioning index, arr[pi] is now at right place */
    pi = partition(arr, low, high);
    quickSort(arr, low, pi - 1);  // Before pi
    quickSort(arr, pi + 1, high); // After pi
  }
}
```

```
/* This function takes last element as pivot, places the pivot element at its correct position in sorted array, and places all
smaller (smaller than pivot) to left of pivot and all greater elements to right of pivot */
partition (arr[], low, high)
{
  // pivot (Element to be placed at right position)
  pivot = arr[high];
  i = (low - 1)  // Index of smaller element and indicates the
  // right position of pivot found so far
  for (j = low; j <= high - 1; j++){
    // If current element is smaller than the pivot
    if (arr[j] < pivot){
      i++;   // increment index of smaller element
      swap arr[i] and arr[j]
    }
  }
  swap arr[i + 1] and arr[high])
  return (i + 1)
}
```

- Δ

Η παραγωγή των δοκιμαστικών δεδομένων για το μέρος Δ έγινε με τη χρήση της κλάσης Random του πακέτου java.util. Συγκεκριμένα δοσμένου ενός πίνακα N ο οποίος περιέχει τα διάφορα ν για τα οποία θα δημιουργήσουμε δεδομένα (οπου ν ειναι το πλήθος φακέλων, ή αλλιώς το πλήθος των γραμμών σε κάθε αρχείο), καθώς και ενός αριθμου amount ο οποίος μας λέει πόσα αρχεία θα παράγουμε για κάθε ν θα πάρουμε τον παρακάτω ψευδοκώδικα:

```
Random r = new Random()

for  n in N

for  I in 0..amount

for j in 0..n

int generated_number = r.nextInt(0, 1_000_001)

file.write(generated_number + "\n")
```

Τα ονόματα για ευκολία δίνονται από τον παρακάτω τύπο:

nNiI όπου N ειναι το πλήθος φακέλων-γραμμών για το αρχείο και το I είναι ο δείκτης του αρχείου για τη συγκεκριμένη κατηγορία

Αυτό έχει ως αποτέλεσμα στη συνάρτηση review η οποία θα τρέξει το benchmark με ενα nested for loop να μπορουμε να φτιάξουμε τα ονόματα όλων των αρχείων και για το κάθε ενα να καλούμε αντίστοιχα τους αλγορίθμους 1 και 2 απο το μέρος β και γ.

Όσον αφορά τα ευρήματα του benchmark για N = 100, 500, 1000 και amount 100 παιρνουμε στα 10 runs ενα μέσο όρο απο 60 δίσκους για N = 100, 300 δίσκους για N = 500 και 600 δίσκους για N = 1000 στον αλγόριθμο 1. Για τον αλγόριθμο 2 οι αριθμοί αυτοί γίνονται 50, 250 και 500 αντίστοιχα.

```
[60, 301, 586]
[53, 264, 507]
```

```
[58, 292, 588]
[52, 256, 509]
```

```
[61, 295, 583]
[55, 259, 506]
```

Η απόδοση των 2 αλγορίθμων παρόλο που το πλήθπς των δεδομένων είναι μικρό, είναι φανερή και υπέρ του αλγορίθμου 2.