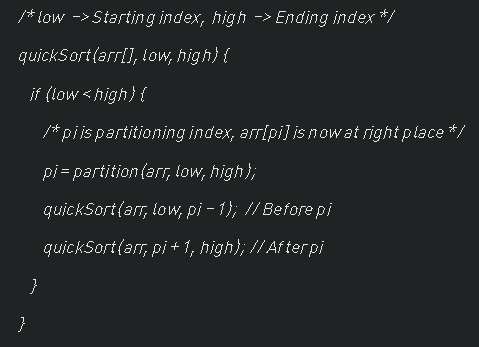
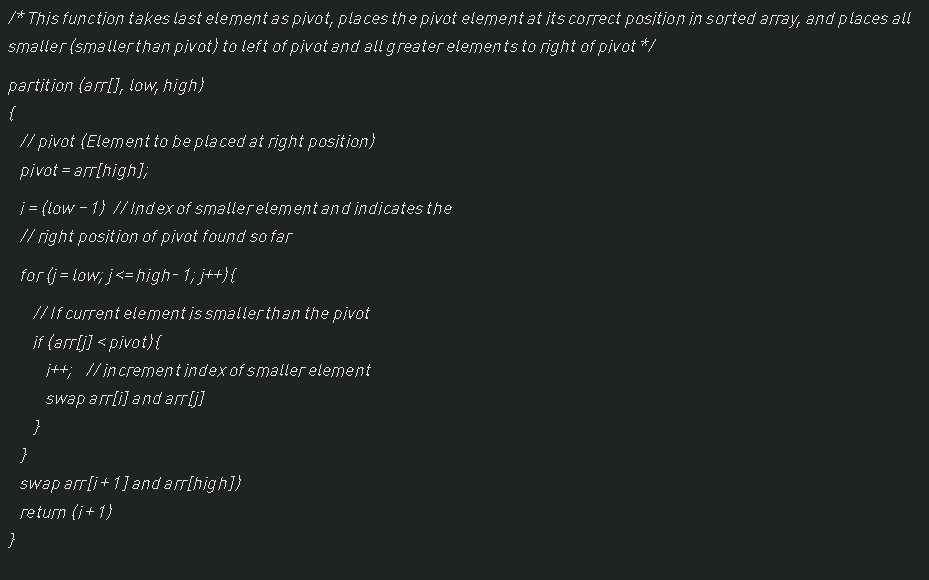
***Αναφορά Παράδοσης (εργασία 2)***

* Α
* Β
* Γ

Για το μέρος Γ χρησιμοποιήθηκε ο αλγόριθμος quick sort.





* 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 Ν 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 Ν ειναι το πλήθος φακέλων-γραμμών για το αρχείο και το Ι 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 Ν = 100, 500, 1000 και amount 100 παιρνουμε στα 10 runs ενα μέσο όρο απο 60 δίσκους για Ν = 100, 300 δίσκους για Ν = 500 και 600 δίσκους για Ν = 1000 στον αλγόριθμο 1. Για τον αλγόριθμο 2 οι αριθμοί αυτοί γίνονται 50, 250 και 500 αντίστοιχα.







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