

## ΑΛΓΟΡΙΘΜΟΙ 2020-2021 – ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΑΣΚΗΣΗ Α' [1 βαθμός]

Μία εταιρεία έχει ένα ανακατωμένο αρχείο. Για κάθε ένα έγγραφο που δεν είναι στη σωστή σειρά χάνει έμμεσα κάποια χρήματα. Η εταιρεία προσλαμβάνει ένα φοιτητή Πληροφορικής που γνωρίζει από Αλγορίθμους για να υπολογίσει το κόστος. Ο εν λόγω φοιτητής προτείνει την παρακάτω ορθή λύση:

«Να σχεδιαστεί και να υλοποιηθεί μία επέκταση της ταξινόμησης συγχώνευσης, η οποία να καταμετράει το κόστος των αντιστροφών. Οι αντιστροφές ορίζονται ως εξής: Έστω ένας πίνακας  $[A_1, A_2, \dots, A_N]$ . Υπάρχει αντιστροφή όταν  $A_i > A_j$ , ενώ  $i < j$ . Για κάθε περίπτωση αντιστροφής το κόστος είναι 3. Στην ειδική περίπτωση όμως όπου  $A_i = A_{j+1}$ , το κόστος είναι 2.»



Ζητείται να υλοποιήσετε τον αλγόριθμο αυτό. Ως είσοδος, θα δίνεται ένα αρχείο στην παρακάτω μορφή:

```
14 9 10 7 3 ...
```

Δηλαδή θα περιέχει τα δεδομένα του πίνακα, τα οποία θα είναι **ακέραιοι μη αρνητικοί** αριθμοί χωρισμένοι μεταξύ τους με ένα κενό. Το πρόγραμμα θα πρέπει να υπολογίζει και να εκτυπώνει στην οθόνη το **συνολικό κόστος** αυστηρά στην εξής μορφή εξόδου:

```
Total cost: 240
```

### Τρόπος υλοποίησης:

- 1) Να χρησιμοποιήσετε αποκλειστικά την γλώσσα **Java**, ακολουθώντας τις οδηγίες για την υποβολή ασκήσεων java στο σύστημα Eagle ([Οδηγίες-Eagle-2021.pdf](#)).
- 2) Το **μοναδικό όρισμα εισόδου** να είναι το αρχείο εισόδου. Το όρισμα αυτό θα πρέπει να είναι το **args[0]** από τα ορίσματα της command line, το οποίο διαβάζεται από την **main**.
- 3) Για τη λύση να χρησιμοποιήσετε αλγόριθμο που να είναι αποδοτικός [ **$O(n \log n)$** ] καθώς θα υπάρχει χρονικός περιορισμός στην εκτέλεση των test cases.
- 4) Ο πηγαίος κώδικας να έχει **συνοπτικά σχόλια** μέσα στον κώδικα (**inline**) και **σχόλια** επάνω από κάθε συνάρτηση, τα οποία να εξηγούν το σκεπτικό της υλοποίησής σας.

### Παραδοτέο:

- Όλο το πρόγραμμα (πηγαίος κώδικας) θα πρέπει να είναι υλοποιημένο σε **ένα και μοναδικό αρχείο java** το οποίο θα έχει όνομα **MergeSort.java** (κλάση MergeSort).
- Το αρχείο αυτό θα πρέπει να το ανεβάσετε και να το **υποβάλετε** στο σύστημα **Eagle** (eagle.csd.auth.gr) με τον λογαριασμό σας.
- Στο σύστημα Eagle θα γίνεται **αυτόματη εκτέλεση** του κώδικά σας σε διάφορα test cases (και φανερά αλλά και κρυφά) καθώς και **αυτόματη βαθμολόγηση** (μέγιστος βαθμός το **100**).
- Μπορείτε να υποβάλετε στο σύστημα μέχρι και 20 φορές. Συνεπώς θα πρέπει να έχετε κάνει την υλοποίηση πρώτα σε ένα άλλο περιβάλλον της επιλογής σας (π.χ. Netbeans, IntelliJ, κλπ.) και να το έχετε δοκιμάσει ώστε **να τρέχει σωστά** και **να παράγει σωστά αποτελέσματα** πριν κάνετε υποβολές.
- Έναρξη υποβολών: **Δευτέρα 10/5/2021 10:00πμ.**
- Τέλος υποβολών: **Πέμπτη 20/5/2021 11:59μμ.**

### Διευκρινίσεις:

- Επιτρέπεται η χρησιμοποίηση έτοιμου κώδικα ελεύθερα διαθέσιμου (έτοιμες υλοποιήσεις με κατάλληλη προσαρμογή από εσάς) με την προϋπόθεση ότι θα αναφέρεται σαφώς στα σχόλια η πηγή του. Όμως, τα προγράμματα θα ελέγχονται από αυτόματο σύστημα εντοπισμού αντιγραφών (αυτό που υπάρχει στο Eagle). Αν εντοπιστούν αντιγραφές μεταξύ φοιτητών, τότε οι φοιτητές θα μηδενίζονται.
- Στο επάνω μέρος του αρχείου java με τον πηγαίο κώδικα να αναφέρονται οπωσδήποτε σε σχόλια το ονοματεπώνυμο, το ΑΕΜ και το ακαδημαϊκό email σας.
- Οι φοιτητές θα πρέπει να είναι έτοιμοι να δώσουν προφορικές εξηγήσεις για την υλοποίησή τους, εφόσον τους ζητηθεί.
- **Οι βαθμοί των ασκήσεων θα ισχύουν και για τις εξετάσεις Σεπτεμβρίου ή επί πτυχίω.**

### Μερικά test cases:

Είσοδος:

Έξοδος:

1 3 4 2 5	Total cost: 5
10 10 9 9 3 3 3 3 1 1	Total cost: 104
1 2 3 4 5 6 7 15 14 13 12 11 10 9 8	Total cost: 77