

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

Η άσκηση αυτή ασχολείται με δύο παραλλαγές του προβλήματος του σακιδίου που διδάχθηκε. Πιο συγκεκριμένα, θεωρούμε ότι έχουμε ένα εμπορευματοκιβώτιο (container) μίας δεδομένης χωρητικότητας. Στη διάθεσή μας έχουμε ένα πλήθος από κουτιά (με εύπλαστο σχήμα) σταθερής χωρητικότητας. Το κάθε είδος κουτιού είναι διαθέσιμο σε διαφορετικές ποσότητες. Θέλουμε να γεμίσουμε πλήρως το εμπορευματοκιβώτιο χρησιμοποιώντας τον μικρότερο αριθμό από τα διαθέσιμα κουτιά. Απαραίτητη προϋπόθεση είναι να μην αφεθεί καθόλου κενός χώρος στο εμπορευματοκιβώτιο.



Το πρόγραμμά μας θα παίρνει και θα διαβάζει (ως μοναδικό όρισμα) ένα αρχείο εισόδου στην παρακάτω μορφή (σε κωδικοποίηση UNIX-ANSI):

```
120
9
1 28 15
2 21 7
...
9 14 27
```

Δηλαδή, στην 1^η γραμμή το αρχείο θα περιέχει την χωρητικότητα του εμπορευματοκιβωτίου, στη 2^η γραμμή θα περιέχει το πλήθος των διαθέσιμων τύπων κουτιών, και στη συνέχεια στις επόμενες γραμμές για κάθε έναν τύπο κουτιού θα δίνεται ο χώρος που καταλαμβάνει στο εμπορευματοκιβώτιο και η ποσότητα στην οποία είναι διαθέσιμο (τα στοιχεία αυτά θα χωρίζονται με κενά).

Στη συνέχεια το πρόγραμμα θα πρέπει να υπολογίζει το **ελάχιστο πλήθος των κουτιών που απαιτούνται** στις παρακάτω δύο περιπτώσεις:

α) θεωρώντας άπειρες ποσότητες κουτιών (αγνοώντας δηλαδή τις διαθέσιμες ποσότητες) [unbounded case], και

β) λαμβάνοντας υπόψη τις ποσότητες που δίνονται [bounded case].

Τα αποτελέσματα θα πρέπει να εκτυπώνονται στην οθόνη αυστηρά στην εξής μορφή εξόδου:

```
The minimum number of boxes for the unbounded case is: 7
The minimum number of boxes for the bounded case is: 12
```

Προσοχή: Δεν είναι αναγκαστικό να υπάρχει λύση για οποιοδήποτε αρχείο εισόδου. Σε αυτή την περίπτωση να βγαίνει το ακόλουθο μήνυμα στην οθόνη (ανάλογα με το ποια περίπτωση δεν έδωσε λύση): «There is no solution for the bounded/unbounded case.»

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 (π.χ. δυναμικού προγραμματισμού) 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 Knapsack.java** (κλάση Knapsack).
- 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 31/5/2021 10:00πμ.**
- Τέλος υποβολών: **Πέμπτη 10/6/2021 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.**

1ο Παράδειγμα εκτέλεσης:

Έστω το αρχείο εισόδου με δεδομένα:

```
200
2
1 11 10
2 23 3
```

Η έξοδος θα είναι:

```
The minimum number of boxes for the unbounded case is: 16
There is no solution for the bounded case.
```

Επεξήγηση:

- Στην unbounded case θα χρησιμοποιήσουμε: **14** κουτιά τύπου-1 και **2** κουτιά τύπου-2, για να πετύχουμε το ελάχιστο. (χωρητικότητα: $14 \cdot 11 + 2 \cdot 23 = 200$)
- Στην bounded case ακόμα και όλα τα διαθέσιμα κουτιά να χρησιμοποιήσουμε δεν φτάνουμε την απαραίτητη χωρητικότητα: $10 \cdot 11 + 3 \cdot 23 = 179 < 200$.

2° Παράδειγμα εκτέλεσης:

Έστω το αρχείο εισόδου με δεδομένα:

```
1100
5
1 24 20
2 46 12
3 78 8
4 21 11
5 30 19
```

Η έξοδος θα είναι:

```
The minimum number of boxes for the unbounded case is: 17
The minimum number of boxes for the bounded case is: 20
```

Επεξήγηση:

- Στην unbounded case θα χρησιμοποιήσουμε: **3** κουτιά τύπου-1, **2** κουτιά τύπου-2, και **12** κουτιά τύπου-3 για να πετύχουμε το ελάχιστο. (χωρητικότητα: $3 \cdot 24 + 2 \cdot 46 + 12 \cdot 78 = 1100$)
 - Στην bounded case θα χρησιμοποιήσουμε: **2** κουτιά τύπου-1, **11** κουτιά τύπου-2, και **7** κουτιά τύπου-3 για να πετύχουμε το ελάχιστο. (χωρητικότητα: $2 \cdot 24 + 11 \cdot 46 + 7 \cdot 78 = 1100$)
- (ενδεχομένως τα ίδια ελάχιστα να επιτυγχάνονται και με διαφορετικές κατανομές)