

1)

ShopSmart.py

Η συνάρτηση εκτελεί ένα βρογχο για κάθε μαγαζί, όπου στην ουσία τσεκάρει αν υπάρχει το φρούτο της παραγγελίας στο μαγαζί . Αν υπάρχει τότε υπολογίζεται ο λογαριασμός , αν δεν υπάρχει μηδενίζεται ο λογαριασμός και προχωράμε στον έλεγχο του επομένου μαγαζιού. Παράλληλα αποθηκεύει τον μικρότερο λογαριασμό και το αντίστοιχο μαγαζί . Τέλος επιστρέφει το όνομα του μαγαζιού.

BuyLotsOfFruits.py

Αν το φρούτο είναι μέσα στη παραγγελία τότε υπολογίζεται ο λογαριασμός αλλιώς μηδενίζεται και δεν συνεχίζεται ο υπολογισμός .

2) Priority Queue

Γενικά η δομή του σωρού αποτελείται από tuples της μορφής (priority , item) συνεπώς όταν θέλουμε να κάνουμε assign μόνο το item , για παράδειγμα, είναι αναγκαστικό να γράφει μια εντολή τύπου [tempPr , tempItem = self.hear[i]] , ενώ αντίστοιχα στην συνάρτηση heappush της βιβλιοθήκης heapq δίνουμε ορισμό το tuple (priority, item)

- **Init** : Ο constructor της κλάσης
- **findItem** : Τσεκάρει αν το item υπάρχει μέσα στο σωρό και αν υπάρχει επιστρέφει τη θέση του αλλιώς επιστρέφει -1
- **Push** : Αν δεν υπάρχει το στοιχείο ήδη μέσα στο σωρό το προσθέτει
- **isEmpty** : boolean που επιστρέφει αν είναι άδειος ή όχι ο σωρός
- **Pop** : Αν δεν είναι άδειος ο σωρός επιστρέφει το item της κορυφής αυτού.
- **Update** : Αρχικά ψαχνουμε να βρούμε αν υπάρχει ήδη το item στον σωρό. Αν δεν υπάρχει τότε πολύ απλά καλούμε την push για το συγκεκριμένο item. Στην περίπτωση που υπάρχει, συγκρινουμε την παλιά priority με την νέα. Αν η παλιά είναι μικρότερη (δηλ μεγαλύτερος αριθμός , αφού το 1 είναι πιο “μπροστά “ στην ουρά από το 8) τότε ενημερώνουμε την προτεραιότητα του . Αν κάτι τέτοιο δεν ισχύει τότε δεν γίνεται καμία αλλαγή
- **HeadPriority** : Χρησιμοποιείται στην PQSort έτσι ώστε να πάρουμε την τιμή της priority της κεφαλής του σωρού , αφού η pop επιστρέφει το item.
- **PQSort** : Δημιουργείται μια μεταβλητή τύπου PriorityQueue και έπειτα γίνεται σε αυτή push ένα ένα τα στοιχεία της λίστας που δώσαμε σαν ορισμό . Έπειτα γίνονται ένα -ένα pop (και λόγω της δομής του σωρού θα βγουν με αυξουσα σειρά) και τα βάζουμε στο τέλος μιας λίστας την οποία και επιστρέφουμε