Δομές Δεδομένων Άσκηση 1

ΝΙΚΟΛΑΣ ΚΟΥΝΤΟΥΡΙΩΤΗΣ 3170195

StringQueueImpl

Για την υλοποίηση του Queue χρησιμοποιήσα Nodes , όπου υπάρχει και private κλάση μέσα στην StringQueueImpl . Κάνει implement το interface που δόθηκε στην εργασία και χρησιμοποιεί Generics <T> για να αποθηκεύει το ανάλογο data μέσα στα Nodes . Για την υλοποίηση αυτής της κλάσης δόθηκαν 6 μεθόδοι :

- isEmpty()
- put(item)
- get()
- peek()
- printQueue(Printstream stream)
- size()

Για να λειτουργεί η λίστα μονής σύνδεσης , πρέπει να έχουμε 2 δείκτες , ένα στο κεφάλι της ουράς και ένα στην ουρά , για να μπορούμε να αφαιρούμε και να προσθέτουμε δεδομένα σε **O(1).** Η μέθοδος size() επιστρέφει μεταβλητή κλάσης που αυξομειώνεται με κάθε put() ή get() , άρα επιστρέφει την τιμή σε **O(1).** Για την υλοποίηση του get απλά δείχνουμε την κεφαλή στον επόμενο Node της υπάρχουσας κεφαλής και έπειτα αφαιρείται από την ουρά . Για την υλοποίηση του put() απλά αλλάζουμε τον δείκτη της ουράς να δείχνει στο καινούργιο στοιχείο , αφού το ενώσουμε με το προυπάρχον τελευταίο στοιχείο.

StringStackImpl

Για την υλοποίηση του StringStackImpl έχουμε παρόμοιες μεθόδους :

- isEmpty()
- push()
- size()
- peek()
- pop()
- printStack()

Με παρόμοιο σκεπτικό με την ουρά προσθέτουμε αντικείμενα της κλάσης Node όπου έχουν ένα τύπο T data και ένα link στον επόμενο Node για να δείχνουν ότι τύπο δεδομένων θέλουμε . Σε ένα stack αφαιρούμε το τελευταίο στοιχείο που προστέθηκε αντιθέτως με την ουρά , άρα στην put και get ενημερώνονται ανάλογα οι τιμές και τα links .

Thiseas

Για την επίλυση αυτού του προβλήματος πρέπει αναγκαστικά να χρησιμοποιηθεί το Stack από το Α' μέρος για backtracking . Πρώτα διαβάζουμε το αρχείο και αποθηκεύουμε τις μεταβλητές για τις διαστάσεις του λαβύρινθου και το σημείο εισόδου ανάλογα σε ένα String Array . Μετά μετατρέπουμε το stringArray σε int Array με την τιμή 'Ε' = 2 . Έχουμε ένα διπλό πίνακα που δείχνει τον λαβύρινθο που διαβάστηκε από το αρχείο και ένα βοηθητικό πίνακα visited για να γνωρίζουμε ποιός κόμβος είναι μέσα στο stack . Αφού γίνουν οι σωστοί έλεγχοι (σημείο εισόδου , διαστάσεις πίνακα) , προχωράμε στην μέθοδο isReachable όπου εκεί γίνεται το κύριο μέρος της άσκησης .Περνάμε σαν παραμέτρους σε αυτή την μέθοδο ένα διπλό πίνακα int array thisArr όπου δείχνει τον λαβύρινθο μας , τα σημεία εισόδου του λαβύρινθου και τις διαστάσεις του πίνακα . Αρχικοποιώντας ένα instance της κλάσης StringStackImpl<Node>, όπου κλάση Node είναι μια private class όπου έχει ο κάθε κόμβος συντεταγμένες ,τιμή (0 ή 1 ή 2) και κατεύθυνση .Εκτελώντας το while(!stack.isEmpty()) , παίρνουμε τα στοιχεία του πρώτου κόμβου (συντεταγμένες , τιμή και κατεύθυνση) , μεταβάλλουμε την κατεύθυνση σε +1 και έπειτα κάνουμε pop και προσθήκη του νέου κόμβου με ανανεωμένη την κατεύθυνση . Μετά ελέγχουμε για κάθε κατεύθυνση αν είναι εντός των ορίων του πίνακα , αν είναι 0 η τιμή του τετραγώνου (προσβάσιμο τετράγωνο σύμφωνα με την άσκηση) και αν είναι true η τιμή της στον πίνακα visited . Αν ισχύουν οι προϋποθέσεις τότε προσθέτουμε αυτό το σημείο στο stack για μετέπειτα εξέταση . Αν το direction φτάσει στο 4 τότε έχουμε εξαντλήσει τις κατευθύνσεις για το συγκεκριμένο node και το κάνουμε pop από το stack για να προχωρήσουμε στο επόμενο . Το πρόγραμμα μας έτσι λειτουργεί με backtracking και εξαντλεί όλες τις επιλογές μέχρι να βρεθεί λύση , αν υπάρχει .

Ο χαρακτήρας 'Ε' που βρίσκεται στο text file πρέπει να είναι λατινικός χαρακτήρας.

StringQueueWithOnePointer

Για την υλοποίηση του Μερους Γ έκανα implement το interface που δόθηκε για την πρώτη άσκηση και δημιούργησα την private κλάση Nodee με την οποία χρησιμοποιούμε Generics πάλι . Για την λύση αυτού του προβλήματος πρέπει να έχουμε ένα δείκτη και μια μεταβλητή size (η οποία αυξομειώνεται αναλόγως) , γι αυτό παίρνουμε τον δείκτη της ουράς και χρησιμοποιούμε κυκλική λίστα . Για να προσθέσουμε ένα στοιχείο απλά αλλάζουμε τον δείκτη της ουράς να δείχνει στο αντικείμενο που μόλις προσθέσαμε , με σύνδεση στον προηγούμενο του και με το πρώτο στοιχείο της ουράς . Για να αφαιρέσουμε ένα στοιχείο, απλά δείχνουμε μέσω του δείκτη ουράς το επόμενο στοιχείο ,δηλαδή το πρώτο στοιχείο, το ανταλλάσσουμε με το επόμενο του και επιστρέφουμε το πρώτο στοιχείο. Η διαδικασία αυτή μαζί με την προσθήκη αντικειμένου στην ουρά γίνεται σε **O(1).**

ΤΕΛΟΣ