ΛΕΠΤΟΜΕΡΗΣ ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ-ΠΡΟΣΟΜΟΙΩΣΗΣ

Οι Κλάσεις

Node

Κλάση για κάθε κόμβο της αλυσίδας. Περιέχει έναν δείκτη που δείχνει τον επόμενο κόμβο και είναι αρχικά null, καθώς και δεδομένα σχετικά με τον κόμβο (index, x, y, sec). Έχει έναν default κατασκευαστή που αρχικοποιεί της τιμές των μεταβλητών που προαναφέρθηκαν, αλλά και έναν κατασκευαστή με ορίσματα που θέτει τιμές στις μεταβλητές της κλάσης Node.

Chain

Κλάση για κάθε αλυσίδα. Περιέχει έναν κλασικό δείκτη τύπου Node* head που δείχνει την αρχή της αλυσίδας , μαζί με την προσθήκη ενός πίνακα(array) τύπου Node* ο οποίος δείχνει στην αρχή κάθε ημέρας. Έχει έναν default κατασκευαστή που αρχικοποιεί τον δείκτη head, και έναν κατασκευαστή με όρισμα την νέα τιμή της head. Ακόμα έχει την συνάρτηση newDay() που θέτει τιμές στον πίνακα day σε μια ορισμένη θέση του d. Η κλάση αυτή διαθέτει τις “βοηθητικές” συναρτήσεις :

- **NodeExists.**

  Λειτουργία :Βρίσκει αν υπάρχει κόμβος σε κάποια θέση.

  Περιγραφή:Είναι τύπου Node*, η οποία δέχεται σαν όρισμα την ακέραια μεταβλητή k που αντιστοιχεί στην θέση(index) κάποιου κόμβου. Στη συνέχεια γίνεται προσπέλαση της αλυσίδας ως το τέλος της λίστας και επιστρέφεται η διεύθυνση στην οποία βρέθηκε η θέση(index) που ψάχνουμε ή null αν δεν βρέθηκε.

- **AppendNode**

  Λειτουργία : Προσθέτει κόμβους στο τέλος της αλυσίδας.

  Περιγραφή : Δεν επιστρέφει κάποια τιμή και δέχεται σαν όρισμα τη διεύθυνση ενός κόμβου n , αν αυτός δεν προϋπάρχει (έλεγχος που γίνεται με τη βοήθεια της NodeExists) τότε αν η λίστα είναι κενή , γίνεται η αρχή της (head) . Αλλιώς αν η λίστα δεν είναι κενή πραγματοποιείται προσπέλαση της λίστας ώσπου να βρεθεί ο τελευταίος κόμβος και να τεθεί στον επόμενο του η διεύθυνση του νέου κόμβου της λίστας.

- **insertNode**

  Λειτουργία : Προσθέτει κόμβο ανάμεσα σε άλλους.

  Περιγραφή : Δεν επιστρέφει κάποια τιμή , δέχεται σαν όρισμα τον ακέραιο αριθμό k ο οποίος αντιστοιχεί στην θέση του κόμβου μετά από τον οποίο θα προστεθεί ένας νέος και την διεύθυνση n τύπου Node* που περιέχει πληροφορίες σχετικά με το νέο κόμβο. Αν λοιπόν υπάρχει κόμβος στη θέση k (το βρίσκει με τη βοήθεια της NodeExists) τότε γίνεται προσπέλαση της λίστας ως τον κόμβο αυτόν και ο νέος κόμβος δείχνει πλέον στον επόμενο του κόμβου της θέσης k , και ο κόμβος της θέσης k δείχνει στον νέο.

- **deleteNode**

  Λειτουργία: Διαγράφει έναν κόμβο κάποιας θέσης από την αλυσίδα .

  Περιγραφή : Δεν επιστρέφει τιμές, δέχεται σαν όρισμα την θέση(position) του κόμβου που θα διαγραφεί . Μετά θέτει την μεταβλητή previousNode ίση με την αρχή της λίστας και την currentPos ίση με τον 2ο κόμβο της λίστας. Ύστερα γίνεται προσπέλαση της λίστας ώσπου να βρεθεί ο κόμβος με τη θέση που ψάχνουμε ,όταν γίνει αυτό η διεύθυνση του αποθηκεύεται στην μεταβλητή temp και η προσπέλαση σταματά, μέχρι να βρεθεί οι μεταβλητές που δείχνουν στον προηγούμενο και τρέχον κόμβο πηγαίνουν επαναληπτικά στον επόμενο. Στη συνέχεια αν ο κόμβος που ψάχνουμε έχει βρεθεί και άρα η temp δεν είναι null ,ο προηγούμενος κόμβος συνδέεται με τον επόμενο κόμβο από αυτόν που βρήκαμε και αυτός που βρήκαμε διαγράφεται.

Η κλάση Chain περιέχει και 2 βασικές συναρτήσεις την REPAIR και την SUMMARIZE_TRAJECTORY.

Οι Συναρτήσεις

REPAIR

Λειτουργία: Επισκευή αλυσίδας με προσθήκη χαμένων κόμβων.

Περιγραφή: Η συνάρτηση αυτή δεν επιστρέφει κάποια τιμή, και δέχεται σαν όρισμα την ημέρα η οποία θα "επισκευαστεί" . Αρχικά εισάγεται η αρχή της λίστας προς επισκευή στην μεταβλητή ptr από τον πίνακα με τις αρχές των ημερών (day[14]). Στη συνέχεια πραγματοποιείται προσπέλαση της λίστας και αν βρεθούν 2 κόμβοι με χρονική διαφορά μεγαλύτερη των 30 δευτερολέπτων ,δηλαδή λείπουν κόμβοι , τότε υπολογίζεται ο αριθμός των κόμβων που πρέπει να προστεθούν , ο οποίος ισούται με την ακέραια διαίρεση της διαφοράς των κόμβων με το 30. Έπειτα η τρέχουσα θέση(index) αποθηκεύεται προσωρινά στην μεταβλητή currentKey και αρχίζει να εκτελείται ένας βρόγχος επανάληψης τόσες φορές όσοι είναι και κόμβοι που πρέπει να προστεθούν . Για αρχή δημιουργείται ένα αντικείμενο τύπου Node* στο οποίο αποθηκεύεται η θέση (η θέση του προηγούμενου κόμβου αυξημένη

κατά 1) , οι συντεταγμένες x,y (((ptr->next->x(αντ.y) - ptr->x(αντ. y) * i / numOfNodesToAdd + ptr->x(αντ. y))
η απόσταση ανάμεσα στους κόμβους με χρονική διαφορά > 30 δευτερολέπτων διά τον αριθμό του κόμβου που προσθέτουμε (πχ 1ος,2ος που προσθέτεται ανάμεσα στους 2 κόμβους με την χρονική διαφορά > 30 δευτερολέπτων) επί το σύνολο των κόμβων που θα προστεθούν + χ (ή αντίστοιχα y)) , η χρονική στιγμή που είναι ο χρόνος του πρώτου από τους κόμβους με τη διαφορά >30 δευτερολέπτων αυξημένη κατά 30 επί την σειρά του κόμβου που προσθέτεται, και μετά ο κόμβος αυτός μαζί με όλη την πληροφορία προσθέτεται στη θέση currentKey με τη βοήθεια της συνάρτησης insertNode. Τέλος η μεταβλητή currentKey γίνεται ίση με τη θέση του κόμβου που μόλις προστέθηκε και η προσπέλαση της λίστας συνεχίζεται.

SUMMARIZE_TRAJECTORY

Λειτουργία: Δημιουργεί μια περίληψη ημερών που απέχουν ένα χρονικό διάστημα από την τρέχουσα ημέρα διαγράφοντας ορισμένους κόμβους.

Περιγραφή: Η συνάρτηση αυτή δέχεται σαν ορίσματα μια ακέραια μεταβλητή d που συμβολίζει την τρέχουσα μέρα και μια ακέραια μεταβλητή dBefore που είναι οι μέρες που πρέπει να περάσουν προκειμένου να γίνει η περίληψη κάποιας ημέρας. Αν λοιπόν έχουν περάσει οι απαραίτητες ημέρες η περίληψη ξεκινά , αρχικά μια μεταβλητή τύπου Node* θέτεται ίση με την διεύθυνση της αρχής της ημέρας που θα γίνει η περίληψη, και μια άλλη ίδιου τύπου δείχνει την αρχή της επόμενης ,δηλαδή το τέλος της διαδικασίας. Υπάρχει ακόμα η μεταβλητή PointOfView που δείχνει τον κόμβο με τον οποίο θα γίνονται οι συγκρίσεις κάθε φορά, και για αρχή είναι ο πρώτος κόμβος της λίστας. Συνεχίζοντας πραγματοποιείται η προσπέλαση της αλυσίδας , και αν η απόσταση του επόμενου κόμβου από τον PointOfView κόμβο είναι εντός της ακτίνας R ο επόμενος κόμβος διαγράφεται με τη βοήθεια της συνάρτησης deleteNode , προσέχοντας πάντα να μην γίνουν υπολογισμοί με κόμβους που δεν υπάρχουν (`if (start->next != end)`), αν η απόσταση τους είναι εκτός της ακτίνας ο επόμενος κόμβος γίνεται ο νέος κόμβος με τον οποίο θα συγκρίνονται οι επόμενοι. Τέλος, πηγαίνουμε στον επόμενο κόμβο.

FIND_CROWED_PLACES

Λειτουργία: Βρίσκει πόσα άτομα πέρασαν από μία περιοχή για τουλάχιστον κάποια ώρα ,μια συγκεκριμένη ώρα.
Περιγραφή: Η συνάρτηση αυτή δέχεται σαν ορίσματα την αρχή του χρονικού διαστήματος στο οποίο θα γίνει η αναζήτηση, το τέλος του τις συντεταγμένες μέσα στις οποίες πρέπει να είναι ο χρήστης , την ελάχιστη χρονική διάρκεια παραμονής του σε αυτές και έναν πίνακα δεικτών που δείχνουν που ήταν ο κάθε χρήστης στην αρχή της συγκεκριμένης ημέρας. Για αρχή το σύνολο των ατόμων που πληρούν τις προϋποθέσεις αυτές είναι μηδέν ,καθώς και η διάρκεια παραμονής του χρήστη εντός του χώρου που έχει ορισθεί. Στη συνέχεια οι χρήστες ελέγχονται και γίνεται προσπέλαση των αλυσίδων τους , αν οι κόμβοι είναι εντός του χρονικού διαστήματος που επέλεξε ο χρήστης , ο έλεγχος συνεχίζεται περαιτέρω και αν ήταν στην τοποθεσία που έχει τεθεί τότε η διάρκεια αυξάνεται κατά 30 , αφού κάθε κόμβος δημιουργείται ανά 30 δευτερόλεπτα, αν η διάρκεια είναι μεγαλύτερη της ελάχιστης παραμονής εντός των συντεταγμένων ο αριθμός των ατόμων αυξάνεται και ο επαναληπτικός έλεγχος για τον χρήστη αυτό σταματά, αν όμως ο χρήστης έφυγε από τον χώρο που έχει ορισθεί η διάρκεια μηδενίζεται. Τέλος η διαδικασία ξαναρχίζει από την αρχή

της αλυσίδας του επόμενου χρήστη, όταν ελεγχθούν όλοι η συνάρτηση επιστρέφει το άθροισμα των ατόμων που πληρούσαν τις προϋποθέσεις που ελέγχθηκαν.

POSSIBLE_COVID_19_INFECTION

Λειτουργία: Βρίσκει άτομα που πιθανόν να έχουν COVID-19.
Περιγραφή: Η συνάρτηση αυτή δέχεται σαν όρισμα την μεβλητή inHealth που δείχνει που ήταν ο υγιής χρήστης στην αρχή της ημέρας, την μεταβλητή inCovid που δείχνει που ήταν ο άρρωστος χρήστης στην αρχή της ημέρας , την user που δείχνει την αλυσίδα ποιού χρήστη ελέγχουμε ,και θα εκτυπωθεί αργότερα αν προκύψει υποψία για κορονοιό και την leftFromYesterday που έχει κρατήσει την διάρκεια που ο χρήστης βρισκόταν ή όχι (αν είναι μηδέν) κοντά στον ασθενή στο τέλος της προηγούμενης ημέρας. Αρχικά η ακέραια μεταβλητή counter παίρνει την τιμή 1 και συμβολίζει σε ποιόν κόμβο βρισκόμαστε στην αλυσίδα του υγιούς χρήστη , η flag που είναι η υποψία για κορονοιό είναι false, και η διάρκεια ίση με ότι ήταν στο τέλος της προηγούμενης ημέρας, έτσι ώστε να συνεχιστεί ο έλεγχος. Έπειτα γίνεται προσπέλαση της αλυσίδας του υγιούς χρήστη , στην αρχή η προσωρινή μεταβλητή tmp γίνεται ίση με τον πρώτο κόμβο της ημέρας του ασθενούς ,και ένας μετρητής counter2 που δείχνει σε ποιόν κόμβο βρισκόμαστε στην αλυσίδα του ασθενούς ,παίρνει την τιμή 1 . Όσο ο κόμβος του ασθενούς δεν ξεπερνά σε θέση αυτόν του υγιούς χρήστη και δεν βρισκόμαστε στο τέλος της αλυσίδας του ασθενούς , τότε συγκρίνουμε τον τρέχον κόμβο του υγιούς χρήστη με όλους τους κόμβους του ασθενούς, αρκεί η χρονική διαφορά των κόμβων δεν ξεπερνά τις 5 ώρες (`5 ώρες 300 λεπτά επί 2 30λεπτα σε κάθε ένα => 300*2 = 600`). Τότε αν η απόσταση αυτή είναι μικρότερη από 400 μέτρα για διάρκεια τουλάχιστον 1,5 λεπτού τότε ο χρήστης πιθανόν να έχει κολλήσει κορονοιό, αλλίως αν στον επόμενο οι υποθέσεις δεν ικανοποιούνται η διάρκεια μηδενίζεται, το υπόλοιπο της διάρκειας αποθηκεύεται ώστε να χρησιμοποιηθεί η τελευταία αποθήκευση την επόμενη μέρα. Στο τέλος κάθε εσωτερικής επανάληψης ο μετρητής της αλυσίδας του ασθενούς αυξάνεται κατά 1 και πηγαίνουμε στον επόμενο κόμβο του ασθενούς, αντίστοιχα στο τέλος κάθε επανάληψης του εξωτερικού βρόγχου ο μετρητής counter αυξάνεται κατά 1 και πηγαίνουμε στον επόμενο κόμβο της αλυσίδας του υγιούς χρήστη. Τέλος αν ο χρήστης είναι πιθανόν μολυσμένος εκτυπώνεται ο αριθμός του στην οθόνη και η συνάρτηση επιστρέφει την τελευταία διάρκεια που αποθηκεύτηκε.

randomMovement

Λειτουργία:Παραγωγή σταθερής τυχαίας κίνησης.
Περιγραφή: Η συνάρτηση δέχεται με αναφορά τα ορίσματα που αντιστοιχούν στις συντεταγμένες του χρήστη έτσι ώστε όποιες αλλαγές γίνουν στην συνάρτηση να διατηρηθούν στην main. Υπάρχει μία μεταβλητή τυχαίας κατάστασης situation που έχει ένα τυχαίο αριθμό (0-13) και ανάλογα με αυτόν τον αριθμό και τη βοήθεια του switch επιλέγεται κάποια κίνηση (case 1-4) ή στασιμότητα (default).Τέλος γίνεται έλεγχος πως η κίνηση είναι εντός του χώρου DxD που έχει επιλεχθεί ,αν όχι τότε αναιρεί την τελευταία κίνηση.

Η Main

Στο πρώτο τμήμα της κύριας συνάρτησης γίνεται η δήλωση των μεταβλητών που θα χρησιμοποιηθούν παρακάτω. Στο δεύτερο τμήμα υπάρχουν 2 βασικοί βρόγχοι επανάληψης ,ένας για της ημέρες λειτουργίας του προγράμματος και στο εσωτερικό του ένας για τους 100 χρήστες , **ο πρώτος εκ των οποίων θεωρείται ασθενής**. Αρχικά , αρχικοποιείται η μεταβλητή seconds που μετρά τα δευτερόλεπτα της ημέρας και αρχίζουν να προσθέτονται επαναληπτικά κόμβοι στην αλυσίδα, κάθε άτομο αρχίζει από μία τυχαία τοποθεσία(μέρα 1η) και συνεχίζει την κίνηση του από εκεί που σταμάτησε την προηγούμενη μέρα καθώς η τοποθεσία του κάθε χρήστη αποθηκεύεται στους πίνακες x11[100] και y11[100]. Για την προσομοίωση της απώλειας του σήματος GPS υπάρχει η μεταβλητή random για παραγωγή τυχαίων αριθμών (1-10) αν είναι ίση με 10 ,τότε δεν δημιουργείται νέος κόμβος , όμως ο χρόνος συνεχίζει να αυξάνεται, αν πάλι είναι από 1-9 τότε γίνεται εισαγωγή δεδομένων στον κόμβο . Ακόμα δημιουργούνται πίνακες που δείχνουν τον κόμβο στην αρχή της συγκεκριμένης ημέρας για κάθε χρήστη, αλλά και συγκεκριμένα και για τον ασθενή,καθώς και μια συνάρτηση που στέλνει τον κόμβο της αρχής κάθε μέρα σε κάθε αλυσίδα (αποθηκεύεται στον πίνακα days[14]). Στην συνέχεια καλείται η REPAIR , η POSSIBLE_COVID_19_INFECTION που επιστρέφει το υπόλοιπο της διάρκειας στον πίνακα leftovers[100] και την πέρνει και σαν όρισμα, η SUMMARIZE_TRAJECTORY που κάνει περίληψη του κομματιού της αλυσίδας που καταγράφτηκε 10 μέρες πριν. Τέλος καλείται η FIND_CROWDED_PLACES η οποία παίρνει δεδομένα από το χρήστη και προσφέρεται η δυνατότητα μόνιμης παράκαμψης της για μία μέρα(εισάγοντας οποιαδήποτε λέξη) ή και ως το τέλος του προγράμματος δίνοντας ως απάντηση τη λέξη “STOP”.