

## Spacedeli

Η παρακάτω λύση σε ψευδοκώδικα είναι απλή προσαρμογή του [αλγόριθμου του Dijkstra](#) για την εύρεση συντομότερου μονοπατιού σε γράφο. (Θα ήταν απλούστερη η προσαρμογή αν είχα αφήσει το `dist`, όπως στον ψευδοκώδικα της wikipedia, όμως για να σας διευκολύνω σας δείχνω ακριβώς πώς να χρησιμοποιήσετε μία κατάλληλη δομή δεδομένων  $Q$ , μέσα στην οποία αποθηκεύονται οι καταστάσεις και το κόστος τους.)

Προσέξτε τα εξής:

- Δε χρειάζεται να δημιουργηθεί κανένας γράφος: ο χάρτης αρκεί.
- Χρειάζεται μία **ουρά προτεραιότητας** ή ένα διατεταγμένο σύνολο για το  $Q$ .
- Οι καταστάσεις που θα τοποθετούνται στο  $Q$  είναι ζεύγη αποτελούμενα από δύο στοιχεία:
  - μία τιμή `true/false` που αντιστοιχεί στο αν ο Λάκης κουβαλάει ή όχι την πίτσα
  - τις συντεταγμένες όπου βρίσκεται ο Λάκης

Χρειάζονται και τα δύο γιατί είναι διαφορετική κατάσταση να βρίσκεται κάπου ο Λάκης και να κουβαλάει την πίτσα, και διαφορετική να την έχει αφήσει.

- Η διάταξη των καταστάσεων στο  $Q$  πρέπει να είναι αύξουσα ως προς το κόστος μετάβασης από την αρχική κατάσταση.

### Ψευδοκώδικας

- `start` = θέση του "S" στο χάρτη
- αρχική κατάσταση, `init` = (`true`, `start`)
- ουρά προτεραιότητας καταστάσεων,  $Q$  = κενή
- βάλτε στην  $Q$  την `init` με προτεραιότητα 0
- πίνακας καταστάσεων που έχουμε επισκεφθεί, `prev` = κενός
- `prev[init]` = "none"
- όσο η  $Q$  δεν είναι άδεια
  - αφαίρεσε την κατάσταση (`loaded`, `i`, `j`) από την  $Q$  με την ελάχιστη προτεραιότητα `cost`
  - αν στη θέση (`i`, `j`) υπάρχει "E" και `loaded` = `true`, τότε
    - \* το βέλτιστο κόστος είναι `cost`
    - \* υπολόγισε και επέστρεψε τη λύση
  - αλλιώς, για κάθε δυνατή κίνηση `m` : { "L", "R", "U", "D", "W" }
    - \* υπολόγισε τη νέα κατάσταση `next`, μετά από αυτή την κίνηση
    - \* αν η `next` δεν υπάρχει ήδη στον πίνακα `prev`
      - `prev[next]` = (`i`, `j`, `m`)
      - υπολόγισε το κόστος `c` που επιφέρει αυτή η κίνηση
      - βάλτε στην  $Q$  την `next` με προτεραιότητα `cost` + `c`

Για την ανακατασκευή της λύσης, όταν φτάσετε στο "E" θα χρειαστεί να διατρέξετε προς τα πίσω την πληροφορία που υπάρχει στον πίνακα `prev`, μέχρις ότου φτάσετε στην αρχική κατάσταση (όπου ο πίνακας `prev` περιέχει "none"). Κάθε στοιχείο του πίνακα `prev` μέχρι τότε σας δίνει (σε αντίστροφη σειρά) την προηγούμενη κατάσταση και την κίνηση που πρέπει να γίνει για να φύγουμε από αυτήν.