

## Castle

Η οδηγία λέει ότι στη κάθε θέση προσθέτεις 1,2,4,8 αν έχει τοίχο στα δυτικά, βόρεια, ανατολικά και νότια αντίστοιχα. Οπότε αν το 1ο bit του αριθμού σε κάθε θέση είναι 1 έχει τοίχο στα δυτικά, αν το 2ο είναι 1 έχει τοίχο στα βόρεια κτλ.

Τρέχουμε μια απλή αναδρομική floodfill(αλγόριθμος γεμίσματος) οι οποίοι μετρά τον αριθμό των components(τα δωμάτια), το μέγεθος του κάθε component(για να βρούμε το πιο μεγάλο δωμάτιο) και κάνει assign κάθε κελί  $(i,j)$  σε ένα component. πχ.  $component[i][j]=1$ ; ενώ τα έχουμε όλα αρχικοποιημένα με -1(no component). Προσέξτε να μην βγαίνεται εκτός των boundaries του κάστρου.

Έπειτα περνούμε ξανά πάνω από κάθε κελί  $(i,j)$  ρίχνοντας 1-1 τους τοίχους γύρω του ελέγχοντας αν ενώνει 2 διαφορετικά components(δωμάτια) και αποθηκεύει το πιο μεγάλο δωμάτιο που δημιουργείτε με τη ρίξη ενός τοίχου.

Ο ζητούμενος τοίχος πρέπει να είναι όσο πιο αριστερά και όσο πιο κάτω γίνεται όποτε ξεκινούμε την αναζήτηση από τα κάτω προς τα πάνω και από τα αριστερά προς το δεξιά(από το κάτω αριστερά κελί). Ελέγχουμε για τοίχο μόνο κάτω(N) και μετά δεξιά(E).

Σημείωση: Δεν χρειάζεται να αποθηκεύουμε τον γράφο με adjacency lists ή matrices. Μπορούμε να “δημιουργούμε” δηλ. από το  $(x,y)$  θα ελέγχουμε τα  $(x+1,y)$   $(x-1,y)$   $(x,y+1)$   $(x,y-1)$ .