

#### search:

Για τους αλγορίθμους αναζήτησης, αξιοποίησα την stack για την DFS, queue για την BFS και priority queue για την A\*. περνούσα κόμβους οι οποίοι στα 2 πρώτα περιείχαν tuples με τις συντεταγμένες και μια λίστα από το μονοπάτι που χρειάζεται για να φτάσει εκεί, στο τρίτο έχω και το κόστος που χρειάζεται.

#### searchAgents:

Η  $h(x)$  που έφτιαξα για το corners μετράει το άθροισμα από την απόσταση από την κοντινότερη γωνία στο σημείο που είμαστε και την κοντινότερη γωνία από αυτή την γωνία το οποίο επαναλαμβάνεται για όλες τις γωνίες που δεν έχουν επισκεφθεί. Για την απόσταση χρησιμοποιώ manhattan distance. Είναι admissible, γιατί η απόσταση είναι μικρότερη από την optimal που υπολογίζει και τους τείχους που θα χρειαστεί να περάσει, άρα είναι  $h(s) \leq c(s)$ , για κάθε θέση. Είναι consistent, γιατί  $h(goal)=0$  και  $h(s) \leq c(s,n)+h(n)$  για κάθε state  $s$  και neighbor  $n$ .

Για το food η Heuristic μου χρησιμοποιεί την mazeDistance για να βρει την πραγματική απόσταση του μακρύτερου φαγητού από τον παίκτη. Έτσι θα επιθυμούμαι πάντα να έχουμε όσο πιο κοντά γίνεται και τα πιο μακρινά φαγητά και αν και παίρνει πολλή ώρα η mazeDistance βρίσκει καλύτερα την απόσταση από το manhattan. Είναι admissible, γιατί η απόσταση είναι μικρότερη από την optimal που υπολογίζει την απόσταση που θα διανύσει από το πρώτο μέχρι το τελευταίο φαγητό, άρα είναι  $h(s) \leq c(s)$ , για κάθε θέση. Είναι consistent, γιατί  $h(goal)=0$  και  $h(s) \leq c(s,n)+h(n)$  για κάθε state  $s$  και neighbor  $n$ .