ΟΝΟΜΑ: ΑΓΓΕΛΟΣ ΝΙΚΟΛΑΟΣ

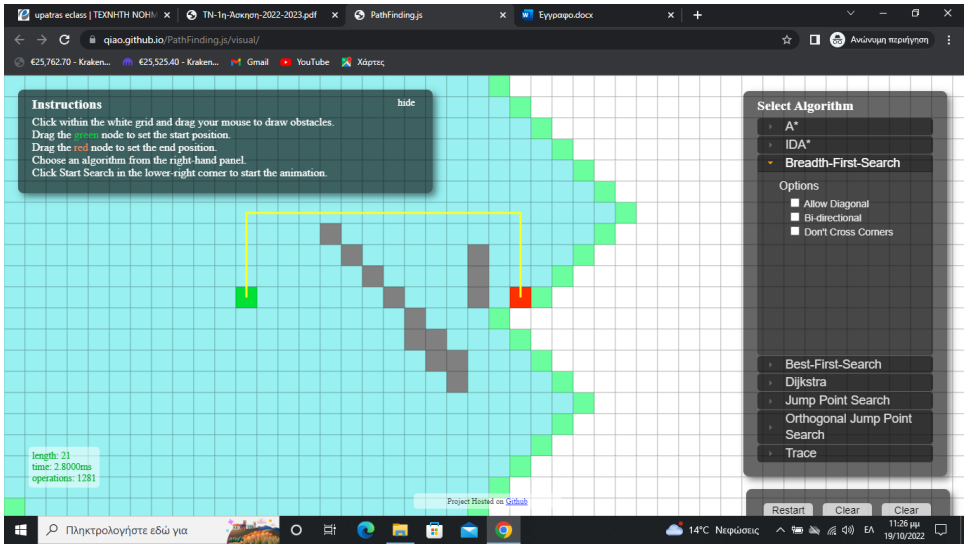
ΕΠΩΝΥΜΟ: ΠΟΤΑΜΙΑΝΟΣ

ΑΜ: 1084537

ΕΤΟΣ ΣΠΟΥΔΩΝ: 3ο

ΜΕΡΟΣ Α:

Το στιγμιότυπό μου:



1)

Αλγόριθμος	Μήκος Λύσης	Είναι βέλτιστη;	Αριθμός βημάτων
BFS	21	ΝΑΙ	1281
Bidirectional	21	ΝΑΙ	743
Dijkstra	21	ΝΑΙ	1308
BestFS, Manhattan	23	ΟΧΙ	80
BestFS, Euclidean	23	ΟΧΙ	76
A*, Manhattan	21	ΝΑΙ	254
A*, Euclidean	21	ΝΑΙ	344

2) Βέλτιστος αλγόριθμος αναζήτησης στην περίπτωση μας, ορίζεται ο αλγόριθμος που μας επιστρέφει την λύση (εφόσον μια κίνηση κατά μήκος των τετράγωνων έχει κόστος ένα) με το μικρότερο κόστος (=βέλτιστη λύση). Όπως παρατηρούμε τόσο από το πινακάκι, όσο και από την θεωρία, όλοι οι αλγόριθμοι μας επιστρέφουν λύση με μήκος 21 εκτός από τον BestFS (και στις δυο παραδοχές του, Manhattan και Euclidean) που μας επιστρέφει λύση με μήκος 23. Αυτό είναι κάτι το οποίο περιμέναμε, αφού ο BestFS χρησιμοποιεί τα άμεσα φθηνότερα βήματα, τα οποία μακροπρόθεσμα δεν είναι αποτελούν πάντα την καλύτερη επιλογή. (Κίνδυνο να μην βρει την βέλτιστη λύση έχει και ο BFS αφού δεν λαμβάνει υπόψιν τα κόστη, ωστόσο στο συγκεκριμένο παράδειγμα την βρήκε αφού τα κόστη είναι ίδια!).

3) Ένας αλγόριθμος αναζήτησης θα λέμε ότι είναι πλήρης εάν (α) σίγουρα μπορεί να εντοπίσει μια (οποιαδήποτε) λύση στον χώρο αναζήτησης εφόσον αυτή υπάρχει. (β) Εάν δεν υπάρχει θα πρέπει να είναι σε θέση να απαντήσει ότι δεν υπάρχει λύση. Επιπροσθέτως από την θεωρία γνωρίζουμε ότι ο BestFS είναι πλήρης εφόσον εκτελούμε αναζήτηση σε διάγραμμα (Graph-Search) αλλά δεν είναι πλήρης όταν εκτελούμε αναζήτηση σε κάποιο δένδρο (Tree Search), αφού υπάρχει περίπτωση να εγκλωβιστεί σε κυκλικά μονοπάτια (loops). Έπεται επομένως πως αν τα κόστη των μονοπατιών δεν είναι ίδια, έχουμε μεγαλύτερη πιθανότητα να ακολουθήσει κάποιο «φθηνότερο» μονοπάτι που θα τον εγκλωβίσει. Στο συγκεκριμένο παράδειγμα ωστόσο που τρέξαμε όλα τα κόστη είναι ίδια και ίσα με την μονάδα, οπότε δεν αντιμετωπίζει κάποιο πρόβλημα. Από την ορισμό της πληρότητας αλγορίθμου όμως καταλαβαίνουμε ότι δεν εκπληρώνονται οι απαιτήσεις (α) και (β) και επομένως μπορεί να θεωρηθεί ο αλγόριθμος ως μη πλήρης.

4) Ως προς την ταχύτητα:

BFS: 1.0000ms
Bidirectional: 0.8000ms
Dijkstra: 1.3000ms
BestFS, Manhattan: 0.1000ms
BestFS, Euclidean: 0.3000ms
A*, Manhattan: 0.3000ms
A*, Euclidean: 0.4000ms

Γενικές Παρατηρήσεις:

I. Ο BestFS, αν και δεν είναι βέλτιστος, υπό περιπτώσεις ούτε καν πλήρης, φαίνεται να μας δίνει την γρηγορότερη λύση (με διαφορά), με την χρήση την ευρετικής Manhattan, καθώς και την λύση (σε ισοπαλία με τον A*) όταν κάνει χρήση της Euclidean ευρετικής, όχι όμως την βέλτιστη λύση (συντομότερο μονοπάτι). Στο συγκεκριμένο παράδειγμα, αν δεν με ενδιέφερε να βρεθεί το συντομότερο μονοπάτι αλλά να βρεθεί γρηγορότερα η λύση, θα τον δεύτερη γρηγορότερη χρησιμοποιούσα.
II. Η Manhattan ευρετική έχει την χρονική πρωτιά στους δύο εξεταζόμενους αλγόριθμους A* (βέλτιστη λύση) και BestFS (μη βέλτιστη αλλά γρηγορότερη λύση), έναντι της Euclidean ευρετικής (Επεξήγηση στο ερώτημα (7) και (8)).
III. Ο πιο χρονοβόρος αλγόριθμος (αν και από τους βέλτιστους) είναι ο UCS (Dijkstra), γεγονός που επιβεβαιώνουμε και από την χρονική του πολυπλοκότητα.

Ανά αλγόριθμος:

I. BFS: Δίνει βέλτιστη λύση (μήκος=21), αλλά κοστίζει αρκετό χρόνο (t=1ms), γεγονός που υποστηρίζεται τόσο από αριθμό των βημάτων που απαιτεί (2ος μεγαλύτερος) όσο και από την θεωρία μας αφού έχει εκθετική χρονική πολυπλοκότητα O(b^d)
II. Bidirectional: Δίνει βέλτιστη λύση (μήκος=21), κοστίζει σχεδόν τον μισό χρόνο (σίγουρα μικρότερο σε κάθε περίπτωση) από τον BFS (με t=0.8ms), γεγονός που υποστηρίζεται τόσο από αριθμό των βημάτων που απαιτεί (3ος μεγαλύτερος) όσο και από την θεωρία μας αφού έχει εκθετική χρονική πολυπλοκότητα O($b^{\frac{d}{2}}+b^{\frac{d}{2}}$)<<O($b^d$)
III. Dijkstra: Δίνει βέλτιστη λύση (μήκος=21), αλλά κοστίζει τον περισσότερο χρόνο (t=1.3ms), γεγονός που υποστηρίζεται τόσο από αριθμό των βημάτων που απαιτεί (1ος μεγαλύτερος) όσο και από την θεωρία μας αφού έχει εκθετική χρονική πολυπλοκότητα) O($b^{1+\lfloor C^*/\varepsilon \rfloor}$).
IV. BestFS Manhattan: Δεν δίνει την βέλτιστη λύση (μήκος=23), κοστίζει όμως τον λιγότερο με διαφορά χρόνο (t=0.1ms), γεγονός που υποστηρίζεται τόσο από αριθμό των βημάτων που απαιτεί (δεύτερα λιγότερα) όσο και από την θεωρία μας αφού έχει χρονική πολυπλοκότητα

O(b^m), η οποία με την χρήση της κατάλληλης ευρετικής (Manhattan), μπορεί να επιφέρει δραματικές βελτιώσεις.

V. BestFS Euclidean: Δεν δίνει βέλτιστη λύση (μήκος=23), κοστίζει όμως τον 2ο λιγότερο με διαφορά χρόνο (t=0.3ms), γεγονός που υποστηρίζεται τόσο από αριθμό των βημάτων που απαιτεί (τα λιγότερα) όσο και από την θεωρία μας αφού έχει χρονική πολυπλοκότητα O(b^m), η οποία με την χρήση της κατάλληλης ευρετικής (Euclidean), μπορεί να επιφέρει δραματικές βελτιώσεις.

VI. A* Manhattan: Δίνει βέλτιστη λύση (μήκος=21), κοστίζει χρόνο (t=0.3ms), γεγονός που υποστηρίζεται τόσο από αριθμό των βημάτων που απαιτεί όσο και από την θεωρία μας αφού έχει εκθετική χρονική πολυπλοκότητα O(b^d). Ο συγκεκριμένος αλγόριθμος είναι πλήρης, βέλτιστος και μας αποδίδει το αποτέλεσμα σε ικανοποιητικό χρόνο.

VII. A* Euclidean: Δίνει βέλτιστη λύση (μήκος=21), κοστίζει χρόνο (t=0.4ms), γεγονός που υποστηρίζεται τόσο από αριθμό των βημάτων που απαιτεί όσο και από την θεωρία μας αφού έχει εκθετική χρονική πολυπλοκότητα O(b^d). Ο συγκεκριμένος αλγόριθμος είναι πλήρης, βέλτιστος και μας αποδίδει το αποτέλεσμα σε ικανοποιητικό χρόνο.

Επομένως ταχύτερος είναι ο BestFS και συγκεκριμένα η παραλλαγή του που κάνει χρήση της ευρετικής Manhattan ενώ και βέλτιστος και ταχύτερος είναι ο A* και συγκεκριμένα η παραλλαγή του που κάνει χρήση της ευρετικής Manhattan.

5) Στον BFS η επέκταση γίνεται δίνοντας προτεραιότητα στο ρηχότερο μονοπάτι ενός κόμβου που δεν έχει ήδη επεκταθεί και οι νέοι κόμβοι προστίθενται στο τέλος (FIFO). Επιστρέφει το βέλτιστο μονοπάτι αν και μόνο αν τα κόστη είναι ίδια.

Στον UCS η επέκταση γίνεται βάση την σειρά του κόστους του μονοπατιού. Βρίσκει πάντα την βέλτιστη λύση.

Έχοντας κατά νου ότι ο UCS αποτελεί παραλλαγή του BFS, το πολύ κοντινό πλήθος βημάτων και χρόνο εκτέλεσης, συμπεραίνουμε ότι στο συγκεκριμένο παράδειγμα η συμπεριφορά τους δεν έχει ουσιαστική διαφορά.

6) Αν και ο «άπληστος» BestFS δεν είναι βέλτιστος και επομένως δεν βρίσκει και την βέλτιστη λύση, βρίσκει όμως την ταχύτερη λύση, κάνοντας χρήση του 1/3 του χρόνου και των αριθμό βημάτων που απαιτεί ο βέλτιστος A*. Ο A* δίνει βέλτιστη λύση, έπεται ότι η ευρετική συνάρτηση h(n)=12 και h*(n)=21 με h(n)≤h*(n) => είναι παραδεκτή και συνεπής και άρα δεν υπερεκτιμά το κόστος. Εν αντιθέσει, ο BestFS επιλέγει μονοπάτι που αποτελείται από άμεσα φθηνότερα βήματα, τα οποία όπως αποδεικνύεται δεν είναι τα βέλτιστα. Επιπροσθέτως γνωρίζουμε ότι ο A* κάνει χρήση των συναρτήσεων των αλγορίθμων BestFS και UCS, έχοντας ως αφετηρία στο δένδρο την τιμή f(x)=h(x)+g(x), όπου h(x)→BestFS και g(x)→UCS. Οπότε αφού χρησιμοποίει αυτές τις δυο παραμέτρους (επιπροσθέτως της συνάρτησης του BestFS δηλαδή), τον καθιστά βέλτιστο σε σχέση με τον BestFS.

7) Το χαρακτηριστικό του συγκεκριμένου στιγμιοτύπου είναι πως δεν επιτρέπει τις διαγώνιες μετακινήσεις.

- Ευρετική Manhattan: Είναι το άθροισμα των οριζόντιων και κάθετων «κινήσεων» που μπορούμε να εκτελέσουμε στο πλέγμα προκειμένου να πάμε από το πράσινο κουτάκι στο κόκκινο κουτάκι (στο συγκεκριμένο παράδειγμα) και δίνεται από το τύπο: Md(π,κ)=$|X\pi - X\kappa| + |Y\pi - Y\kappa|$ και αφού τα Υ είναι ίδια στο συγκεκριμένο παράδειγμα έχουμε: Md(π,κ)=|Χπ-Χκ|

- Ευρετική Euclidean: Μας δίνει την της ευθείας που ενώνει το κόκκινο με το πράσινο κουτί και δίνεται από τον τύπο: d(π,κ)=$\sqrt[2]{(X\pi - X\kappa)^2 + (Y\pi - Y\kappa)^2}$, η υποτείνουσα κοινώς και συγκεκριμένα το ευθύγραμμο τμήμα που ενώνει τα δυο αυτά σημεία. Στο παράδειγμα μας τα Y είναι ίδια οπότε ισούται με: d(π,κ)=$|X\pi - X\kappa|$.

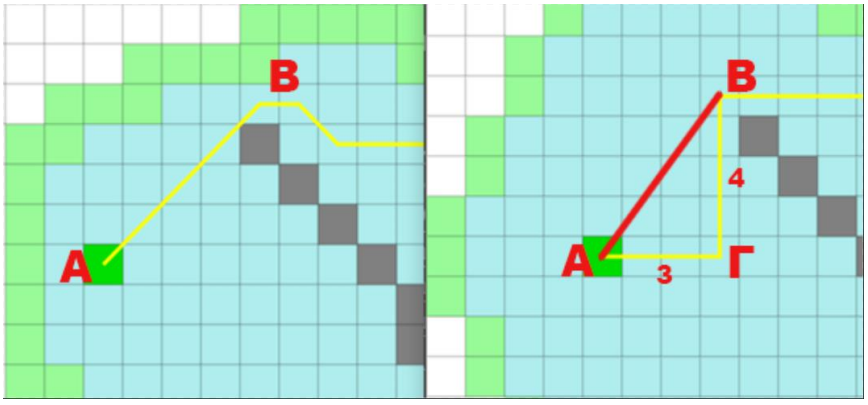
Αν και παίρνουμε το ίδιο τελικό αποτέλεσμα στις δύο αυτές εξισώσεις, με μικρή διαφορά, όταν δεν επιτρέπονται οι διαγώνιες κινήσεις, φαίνεται να είναι ταχύτερη η ευρετική Manhattan αφού η ευρετική Euclidean απαιτεί περισσότερα βήματα κάθε φορά ώστε να καταλήξει στο ίδιο αποτέλεσμα με την Manhattan.

8) Επιτρέποντας την διαγώνια κίνηση στο πλέγμα, θα αξιοποιήσουμε την Euclidean ευρετικη.

Αλγόριθμος	Μήκος Λύσης	Είναι βέλτιστη;	Αριθμός βημάτων	Χρόνος(ms)
BestFS, Manhattan	18.31	Όχι	76	0.5
BestFS, Euclidean	18.9	Όχι	74	0.7
A*, Manhattan	17.14	Όχι	188	2
A*, Euclidean	16.9	Ναι	252	1

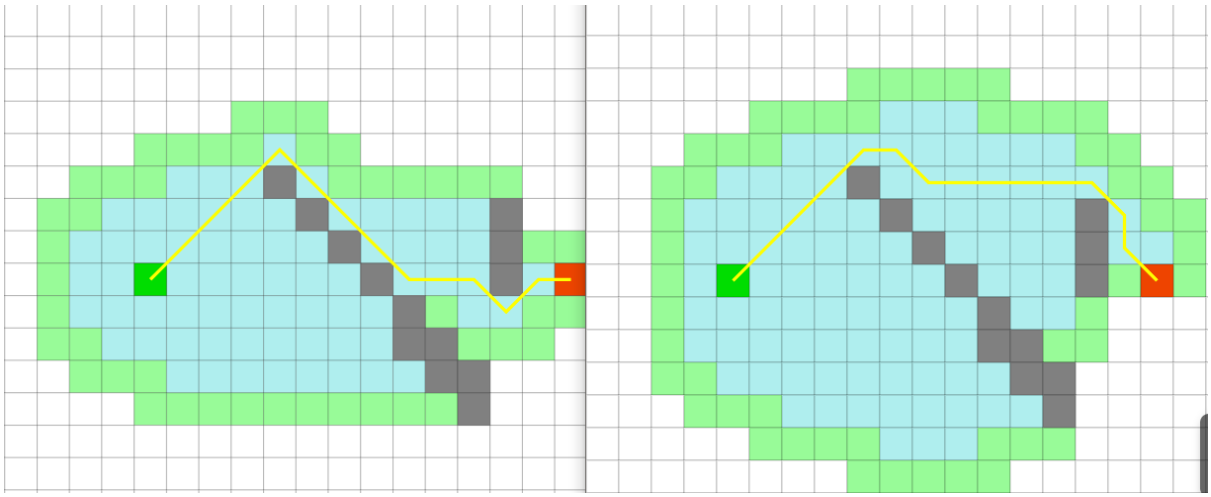
6) Αν και ο «άπληστος» BestFS δεν είναι βέλτιστος και επομένως δεν βρίσκει και την βέλτιστη λύση, βρίσκει όμως την ταχύτερη λύση, κάνοντας χρήση του 1/4 του χρόνου στην ευρετική Manhattan και των 7/10 του χρόνου στην ευρετική Euclidean που απαιτεί ο βέλτιστος αλγόριθμος A*. Ο A* δίνει βέλτιστη λύση μόνο με την ευρετική Euclidean, έπεται ότι η ευρετική συνάρτηση h(n) είναι παραδεκτή και συνεπής και άρα δεν υπερεκτιμά το κόστος. Εν αντιθέσει, ο BestFS ακόμα και με την επιτρεπτή χρήση διαγώνιας διαδρομής, επιλέγει μονοπάτι που αποτελείται από άμεσα φθηνότερα βήματα, τα οποία όπως αποδεικνύεται δεν είναι τα βέλτιστα. Έχει το προβάδισμα ωστόσο στην εκτέλεση συντριπτικά λιγότερων βημάτων καθιστώντας την διαδικασία επιλογής μονοπατιού αισθητά ταχύτερη. Συμπερασματικά, δεν αλλάζει το συμπέρασμα που είχαμε βγάλει με την απαγόρευση της διαγώνιας προσπέλασης.

7) Παρατηρούμε ότι σε αντίθεση με την απαγόρευση κίνησης σε διαγώνιο μονοπάτι, η χρήση της μειώνει το μήκος της λύσης, ωστόσο αυξάνει αισθητά τον απαιτούμενο χρόνο.
Μείωση μήκος λύσης: Όπως φαίνεται προκειμένου να πάει από το A -> B έχει 2 επιλογές. Η Χρήση της διαγώνιας προσπέλασης μας γλιτώνει 2 από το μήκος κίνησης αφού στην αριστερή εικόνα χρειάζεται μήκος 5 (με την χρήση διαγώνιας) ενώ στην δεξιά εικόνα θα χρειαζόμασταν 7 μήκος κίνησης (προφανές).



Εικόνα 1: Στιγμιότυπο A Euclidean diagonal enabled VS A* Euclidean diagonal disabled*

Διαφορά Manhattan και Euclidean: Κατά ορισμό, η Euclidean ευρετική κάνει καλύτερη αξιοποίηση των διαγώνιων αποστάσεων, μιας και αρχή λειτουργίας της βασίζεται στην εύρεση αποστάσεων μεταξύ δυο σημείων βάση της διαγώνιου τους. Η Manhattan φαίνεται να προσπαθεί να βρίσκεται όσο πιο κοντά στον κόμβο-στόχο (κόκκινο κουτί), χωρίς να «υπολογίζει» τα μεταξύ τους εμπόδια. Από τον τρόπο που ορίσαμε τις δυο αυτές ευρετικές, καταλαβαίνουμε ότι η Manhattan είναι ιδανική όταν απαγορεύεται η διαγώνια προσπέλαση ενώ η Euclidean αξιοποιείται πλήρως όταν είναι η επιτρεπτή η διαγώνια προσπέλαση. Οι διαφορές στις δυο αυτές ευρετικές είναι αντίστοιχες για τους δύο διαφορετικούς αλγόριθμους. Η ποιότητα της λύσης επηρεάζεται ωστόσο ανάλογα τον αλγόριθμο, καθιστώντας ως βέλτιστο μόνο τον A* με Manhattan, ενώ χωρίς την διαγώνιων και οι δυο ευρετικές στον A* θεωρούνται βέλτιστες αφού και οι δύο συναρτήσεις σε αυτή την περίπτωση είναι παραδεκτές. Σέ όλες τις περιπτώσεις επηρεάζεται αρνητικά η ταχύτητα των δυο αλγορίθμων σε αντίθεση με την απαγόρευση της διαγώνιας προσπέλασης.



Εικόνα 2: (a) A Manhattan, (b) A* Euclidean*

ΜΕΡΟΣ Β:

1)Το συγκεκριμένο πρόβλημα-πρόγραμμα τρέχει κάνοντας χρήση δύο κατηγοριών αλγορίθμων:

(α) των ευρετικών/κατασκευαστικών αλγορίθμων (Nearest neighbor, Simulated Annealing, βελτιστοποιημένη ευρετική 2-Opt Inversion) οι οποίοι προσπαθούν να βρουν την πιο κοντά στην βέλτιστη λύση, αν δεν καταφέρουν να βρουν την βέλτιστη λύση, σε εύλογο χρονικό διάστημα και (β) των αλγορίθμων εξαντλητικής αναζήτησης (DFS*, Random), οι οποία πάντα θα βρίσκουν την καλύτερη δυνατή λύση αφού «εξαντλούν» κάθε πιθανό μονοπάτι. Αυτό φυσικά σημαίνει πως αν αναφερόμαστε σε μεγάλους χώρους αναζήτησης, οι αλγόριθμοι αυτοί μπορεί να μην καταφέρουν να τερματίσουν σε ρεαλιστικό χρόνο (θα τρέχουν ατέρμονα αν βάλουμε 100 πόλεις π.χ, καθιστώντας δαιδαλώδες το πρόβλημα μας).

Από τον ορισμό που δώσαμε, καταλαβαίνουμε ότι οι αλγόριθμοι που εγγυημένα θα βρουν την βέλτιστη λύση (εφόσον ο χώρος αναζήτησης το επιτρέπει) είναι οι DFS* και Random.

(*)Αναφέρομαι στην bruteforce εκδοχή/ευρετική του αλγορίθμου, αφού γνωρίζουμε οτι ο απλός DFS δεν είναι βέλτιστος, αφού επιλέγει την βαθύτερη λύση πρώτα.

2) Ένας αλγόριθμος αναζήτησης θα λέμε ότι είναι πλήρης εάν (α) σίγουρα μπορεί να εντοπίσει μια (οποιαδήποτε) λύση στον χώρο αναζήτησης εφόσον αυτή υπάρχει. (β) Εάν δεν υπάρχει θα πρέπει να είναι σε θέση να απαντήσει ότι δεν υπάρχει λύση.

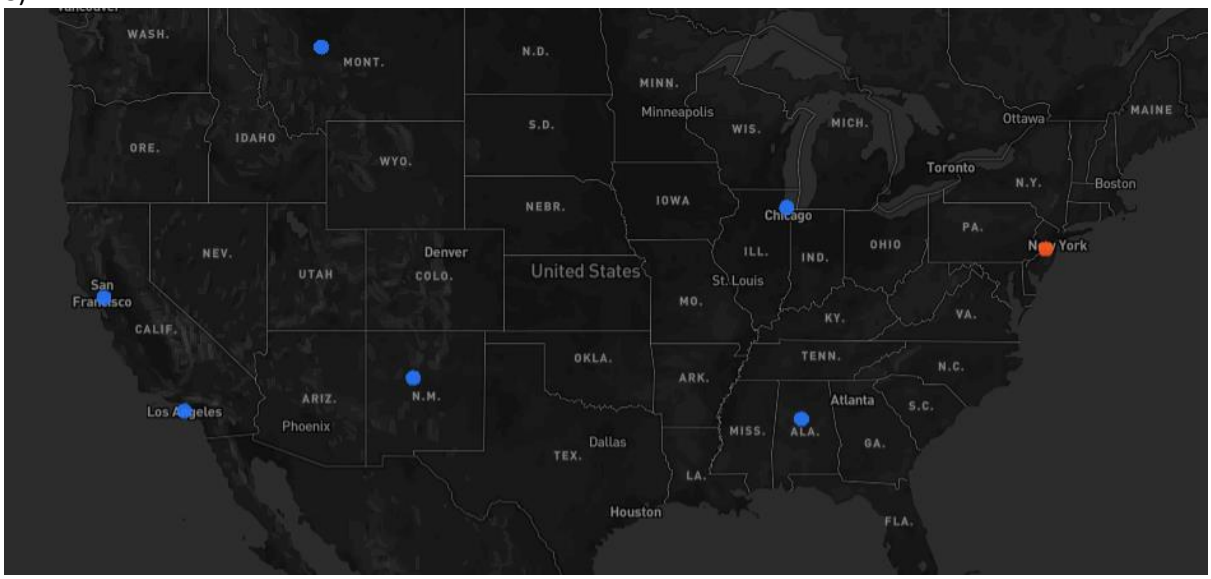
Nearest neighbor: Παρόμοιος με τον BestFS, κληρονομόντας επομένως τις ιδιοτητες του και εφοσον εκτελουμε graph search, ο συγκεκριμενος αλγόριθμος είναι βέλτιστος.

Simulated Annealing: Ο συγκεκριμένος αλγόριθμος αποτελεί μεθευρετική μέθοδοι, οι οποίες χρησιμοποιούνται για την επίλυση γενικευμένων κατηγοριών προβλημάτων, πιο συγκεκριμένα για βελτιστοποιήση των αποτελεσμάτων, υστερόντας ωστόσο στην πληρότητα του.

DFS: Είναι πλήρης αφού στο τέλος θα επεκτείνει όλους τους κόμβους.

Random: Είναι πλήρης αφού θα εξετάσει όλες τις λύσεις.

3)



Αριθμός πόλεων: 7 Σύνολο διαδρομών: 3.600 x 10^2					
Αλγόριθμος	**Μήκος Λύσης**	**Είναι βέλτιστη;**	**Ποσοστό απόκλισης**	**Χρόνος**	**Μήκος/Χρόνος**
Nearest neighbor	9965.61km	ΟΧΙ	~7.1%	<1s	~9965.61
Two-opt inversion	9303.64km	ΝΑΙ	-	6s	1550.6
Simulated Annealing	9303.64km	ΝΑΙ	-	2s	4651.82
DFS	9303.64km	ΝΑΙ	-	468s	20.31
Random	9303.64km	ΝΑΙ	-	12240s+	0.760101

5) Α) i) Όλοι οι αλγόριθμοι βρλισκουν την βέλτιστη λύση εκτος απο τον NN, γεγονός που οφείλεται στην απλειστότητα του αλγορίθμου. Αν και ο NN δεν βρίσκει την καλύτερη λύση, την βρίσκει άμεσα, και γρηγορότερα απο όλους του υπόλοιπους αλγορίθμους (στο συγκεκριμένο παράδειγμα την βρήκε ακαριαία). Από τους αλγόριθμους τώρα, που βρίσκουν την βέλτιστη λύση, γρηγορότερα ανταποκρίθηκε ο Simulated Annealing, μετά η ευρετική Two-opt inversion (στις 10 πόλεις ωστόσο μου βγήκαν αντιμεταθετιμένοι οι δύο αυτοί αλγόριθμοι, κάτι που περίμενα αφού ο Simulated Annealing είναι αποδοτικός και βέλτιστος μόνο σε μικρά

προβλήματα, καθώς όσο αυξάνεται το μέγεθος του προβλήματος, αυξάνεται και ο χρόνος που χρειάζεται). Τέλος, περισσότερη ώρα χρειάστηκαν οι δυο βέλτιστοι αλγόριθμοι, που προφανώς βρήκαν και την βέλτιστη λύση, ο DFS ο οποιος τερμάτισε και ο Random, ο οποιος μεν δεν τερμάτησε, ωστόσο βρήκε την βέλτιστη λύση σχετικά σύντομα (κατα τύχη ωστόσο).
ii) Ο NN αν και βρήκε ακαριαία την λύση, είχε απόκλιση της τάξεων του 7.1%. Οι υπόλοιποι αλγόριθμοι δεν είχαν ποσοστό απόκλισης αφού βρήκαν την βέλτιστη λύση.
Β) Τον καλύτερο λόγο απόδοσης τον έχει ο αλγόριθμος NN, με τιμή 9965.61 (το μεγαλύτερο κλάσμα). Λογικό αφου βρήκε κατευθειαν την διαδρομή, κάνοντας τον διαιρετη≤1.