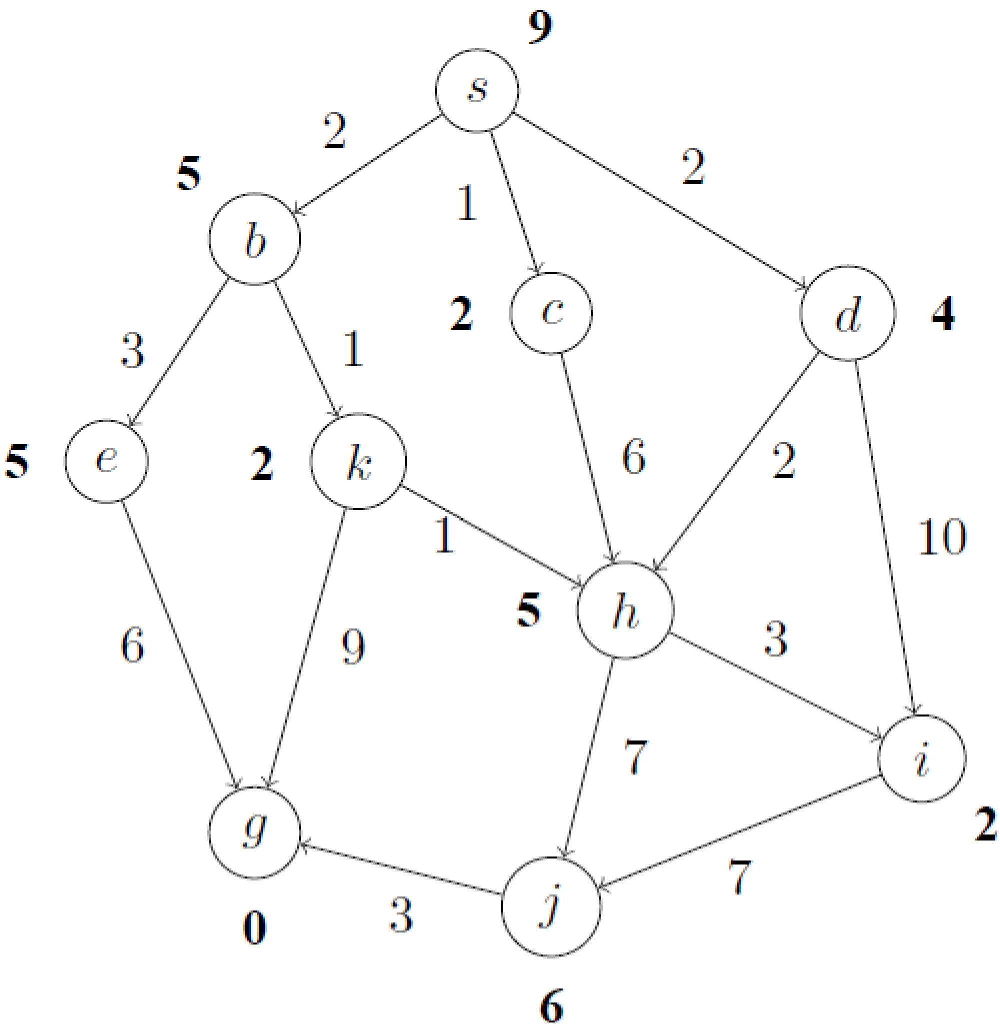
ΜΙΧΑΛΙΤΣΗΣ ΑΛΚΙΒΙΑΔΗΣ ΠΑΝΑΓΙΩΤΗΣ
ΑΜ: 03118868
7Ο ΕΞΑΜΗΝΟ

1Η ΣΕΙΡΑ ΑΣΚΗΣΕΩΝ ΤΟΥ ΜΑΘΗΜΑΤΟΣ «ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ»

Άσκηση 1



1)

<u>Hill Climbing</u>

Βήματα επίλυσης με αλγόριθμο Hill Climbing

Βήμα 1: Ορίζουμε τον τρέχοντα κόμβο ως τη ρίζα του δέντρου.

Βήμα 2: Μέχρι που ο τρέχων κόμβος δεν είναι κόμβος στόχος, εκτελούμε τα ακόλουθα υπο-βήματα:

- **Βήμα 2.α**: Βρίσκουμε τα παιδιά του τρέχοντος κόμβου, και στη συνέχεια βρίσκουμε αυτό με την ελάχιστη υπολογιζόμενη υπόλοιπη απόσταση από το στόχο.
- **Βήμα 2.β**: Εάν ο τρέχων κόμβος δεν έχει παιδιά ή το παιδί που βρέθηκε στο **βήμα 2.α** έχει μεγαλύτερη τιμή ευριστικής από αυτόν πηγαίνουμε στο **Βήμα 3**.
- **Βήμα 2.γ**: Ορίζουμε τον κόμβο που βρέθηκε στο **Βήμα 2.α** ως τρέχων κόμβο.

Βήμα 3: Εάν βρήκαμε ένα κόμβο στόχο τότε ανακοινώνουμε επιτυχία αλλιώς ανακοινώνουμε αποτυχία.

Οπότε αντίστοιχα στον παραπάνω χώρο αναζήτησης έχουμε:

Ξεκινάμε με την αρχική μας κατάσταση s και με βάση τα παραπάνω βήματα ακολουθούμε την παρακάτω διαδικασία:

	HEURISTIC	CHILDREN	WEIGHT C1	WEIGHT C2	WEIGHT C3
S	9	B,C,D	2	1	2
B	5	E,K	3	1	-
C	2	H	6	-	-
D	4	H,I	2	10	-
E	5	G	6	-	-
K	2	G,H	9	5	-
G	0	-	-	-	-
H	5	J,I	7	3	-
I	2	J	7	-	-
J	6	G	3	-	-

(C1,C2,C3 Τα παιδιά από αριστερά προς δεξιά)

Τρέχουσα Κατάσταση	Παιδιά Τρέχουσας Κατάστασης	Μέτωπο Αναζήτησης	Κλειστό Σύνολο
[s]	[c:2,d:4,b:5]	$[(s,9)^{(s)}]$	[]
[c]	[h:5]	$[(c,2)^{(s-->c)}]$	[s]

Επειδή η ευρετική του h είναι μεγαλύτερη από την ευρετική του c, ο αλγόριθμος πάει στο βήμα 3. Και εφόσον το h δεν είναι κόμβος στόχος, ο αλγόριθμος αποτυγχάνει.

Best First
Βήματα επίλυσης με αλγόριθμο Best First

Βήμα 1: Κατασκευάζουμε μια λίστα που περιέχει τη ρίζα του δέντρου (αρχική κατάσταση)
Βήμα 2: Μέχρι που η λίστα να αδειάσει ή να βρεθεί ένας τελικός κόμβος στόχος, εξετάζουμε εάν ο πρώτος κόμβος στη λίστα είναι κόμβος στόχος

- **Βήμα 2.α**: Εάν ο πρώτος κόμβος στη λίστα είναι κόμβος στόχος τότε πηγαίνουμε στο **Βήμα 3**
- **Βήμα 2.β**: Εάν ο πρώτος κόμβος στη λίστα δεν είναι κόμβος στόχος, τότε βγάζουμε το πρώτο κόμβο από τη λίστα, βρίσκουμε στο παιδιά αυτού του κόμβου, βάζουμε στη λίστα τα παιδιά αυτού του κόμβου, και μετά ταξινομούμε σε αύξουσα σειρά ολόκληρη τη λίστα σε σχέση με την ευριστική τιμή για την απόσταση του κάθε κόμβου στη λίστα από ένα κόμβο «στόχο»
- **Βήμα 2.γ**: Εάν ο πρώτος κόμβος δεν έχει παιδιά απλά τον αφαιρούμε από τη λίστα και πηγαίνουμε στο βήμα 2

Βήμα 3: Εάν βρήκαμε ένα κόμβο στόχο τότε ανακοινώνουμε επιτυχία αλλιώς ανακοινώνουμε αποτυχία.

Οπότε αντίστοιχα στον παραπάνω χώρο αναζήτησης έχουμε:

	HEURISTIC	CHILDREN	WEIGHT C1	WEIGHT C2	WEIGHT C3
S	9	B,C,D	2	1	2
B	5	E,K	3	1	-
C	2	H	6	-	-
D	4	H,I	2	10	-
E	5	G	6	-	-
K	2	G,H	9	5	-
G	0	-	-	-	-
H	5	J,I	7	3	-
I	2	J	7	-	-
J	6	G	3	-	-

(C1,C2,C3 Τα παιδιά από αριστερά προς δεξιά)

Τρέχουσα Κατάσταση	Παιδιά Τρέχουσας Κατάστασης	Μέτωπο Αναζήτησης	Κλειστό Σύνολο
[s]	[c:2,d:4,b:5]	[(s,9)$^{\text{(s)}}$]	[]
[c]	[h:5]	[(c,2)$^{\text{(s->c)}}$,(d,4)$^{\text{(s->d)}}$, (b,5)$^{\text{(s->b)}}$]	[s]
[d]	[i:2,h:5]	[(d,4)$^{\text{(s->d)}}$,(b,5)$^{\text{(s->b)}}$,(h,5)$^{\text{(s->c->h)}}$]	[s,c]
[i]	[j:6]	[(i,2)$^{\text{(s->d->i)}}$,(b,5)$^{\text{(s->b)}}$, (h,5)$^{\text{(s->c->h)(s->d->h)}}$]	[s,c,d]
[b]	[k:2,e:5]	[(b,5)$^{\text{(s->b)}}$,(h,5)$^{\text{(s->c- >h)(s->d->h)}}$, (j,6)$^{\text{(s->d->i- >j)}}$]	[s,c,d,i]
[k]	[g:0,h:5]	[(k,2)$^{\text{(s->b->k)}}$, (h,5)$^{\text{(s- >c->h)(s->d->h)}}$, (e,5)$^{\text{(s->b- >e)}}$,(j,6)$^{\text{(s->d->i->j)}}$]	[s,c,d,i,b]
[g]	[]	[(g,0)$^{\text{(s->b->k->g)}}$, (h,5)^$^{\text{(s->c->h)(s->d->h)(s- >b->k->h)}}$, (e,5)$^{\text{(s->b- >e)}}$,(j,6)$^{\text{(s->d->i->j)}}$]	[s,c,d,i,b,k]
Τέλος [g]			

Το μονοπάτι που βρήκαμε είναι το s->b->k->g. Είναι λύση, αλλά όχι η βέλτιστη,

<u>A*</u>

Βήματα επίλυσης με αλγόριθμο A*

- Παρόμοια λογική με του αλγορίθμου Branch and Bound
- Επεκτείνουμε το μονοπάτι με τον καλύτερο από όλους τους κόμβους που βρίσκονται στο μέτωπο αναζήτησης του δένδρου.
- Χρησιμοποιούμε τη σύνθετη ευρετική συνάρτηση F(k) = g(k) + h(k) όπου g(k) είναι η πραγματική απόσταση της k από την αρχική κατάσταση, ενώ υπάρχει και η h(k), μία εκτίμηση της απόστασης της k από το στόχο (μέσω μιας ευρετικής συνάρτησης)
- Εφαρμόζουμε bound

Τρέχουσα Κατάσταση	Παιδιά Τρέχουσας Κατάστασης	Μέτωπο Αναζήτησης	Κλειστό Σύνολο
[s]	[b,c,d]	[(s,0;9)$^{(s)}$]	[]
[c]	[h]	[(c,1;3)$^{(s->c)}$,(d,2;6)$^{(s- >d)}$, (b,2;7)$^{(s->b)}$]	[s]
[d]	[h,i]	[(d,2;6)$^{(s->d)}$,(b,2;7)$^{(s- >b)}$,(h,7;12)$^{(s->c->h)}$]	[s,c]
[b]	[k,e]	[(b,2;7)$^{(s->b)}$,(h,4;9)$^{(s- >d->h)}$, (i,12;14)$^{(s->d->i)}$]	[s,c,d]
[k]	[g,h]	[(k,3;5)$^{(s->b- >k)}$,(h,4;9)$^{(s->d->h)}$, (e,5;10)$^{(s->d->e)}$, (g,12;12)$^{(s->b->k->g)}$, (i,12;14)$^{(s->d->i)}$]	[s,c,d,b]
[h]	[I,j]	[(h,4;9)$^{(s->d- >h)}$, (h,4;9)$^{(s->b->k- >h)}$,(e,5;10)$^{(s->d->e)}$, (g,12;12)$^{(s->b->k->g)}$, (i,12;14)$^{(s->d->i)}$]	[s,c,d,b,k]
[i]	[j]	[(i,7;9)$^{(s->d->h->i)(s->b->k->h->i)}$, (e,5;10)$^{(s->b->e)}$, (g,12;12)$^{(s->b->k->g)}$, (j,11;17)$^{(s->d->h->j)(s->b- >k->h->j)}$]	[s,c,d,b,k,h]
[e]	[g]	[(e,5;10)$^{(s->b->e)}$, (g,12;12)$^{(s->b->k->g)}$, (j,11;17)$^{(s->d->h->j)(s->b- >k->h->j)}$]	[s,c,d,b,k,h,i]
[g]	[]	[(g,11;11)$^{(s->b->e->g)}$, (j,11;17)$^{(s->d->h->j)(s->b- >k->h->j)}$]	[s,c,d,b,k,h,I,e]

Βρήκαμε την λύση s->b->e->g

Ενα από τα χαρακτηριστικά του A* είναι ότι βρίσκει την βέλτιστη λύση.

2)
Στο συγκεκριμένο πρόβλημα προέκυψαν οι εξής λύσεις:

s->b->e->g
s->b->k->g
s->b->k->h->j-> g
s->b->k->h->i->j->g
s->c->h->j->g
s->c->h->i->j->g
s->d->h->j->g
s->d->h->i->j->g
s->d->i->j->g

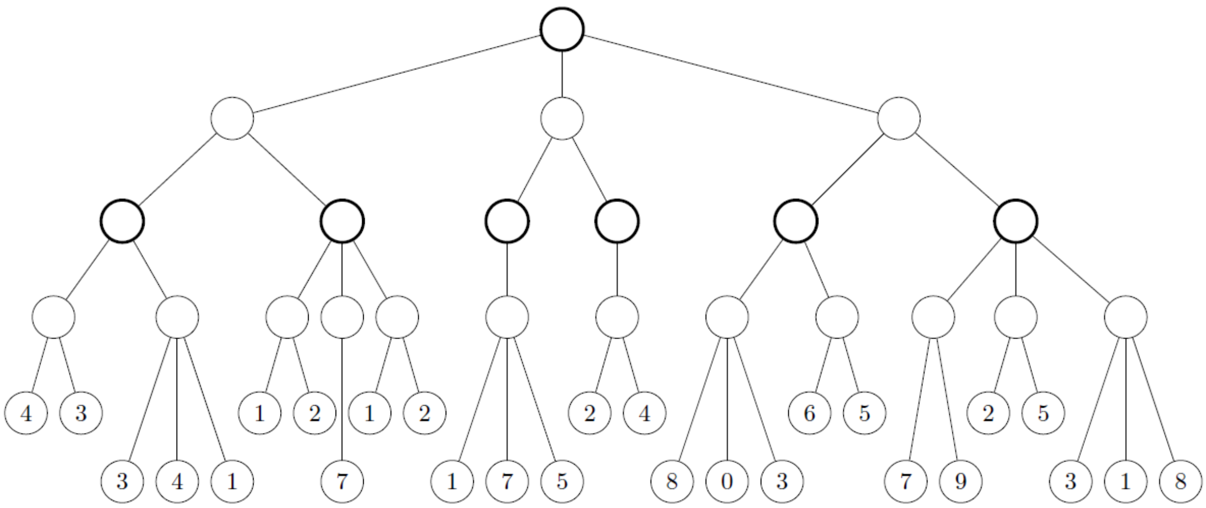
Συνολικά βρήκαμε 9 λύσεις με την χρήση των διαφορετικών αλγορίθμων.
Αναλυτικά:

Χρησιμοποιώντας τον αλγόριθμο Best First βρήκαμε μία λύση και το πρόγραμμα τελείωσε σύμφωνα με την υλοποίηση του αλγορίθμου. Το μονοπάτι της λύσης που βρήκαμε ήταν το s->b->k->g, όπου με το που το βρήκαμε σταματήσαμε εκεί την αναζήτηση.
Ωσόσο όπως είπαμε και παραπάνω ο A* συνήθως βρίσκει την βέλτιστη λύση στα περισσότερα προβλήματα, και το συγκεκριμένο δεν είναι εξαίρεση καθώς η λύση που βρήκαμε με τον Best First εχει κόστος διάσχισης (το ονομάζω BFC) BFC=2+1+9=12.
Αντιθέτως το αντίστοιχο(βέλτιστο) μονοπάτι που βρήκαμε με την υλοποίηση του A* προκύπτει να έχει κόστος διάσχισης(το ονομάζω ASC) ASC=2+3+6=11.
Άρα προφανώς εφόσον ASC<BFC προκύπτει ότι η βέλτιστη λύση του A* με το μονοπάτι s->b->e->g , είναι γενικά η βέλτιστη λύση του προβλήματος.

O Best First είναι ένας αλγόριθμος που δεν εστιάζει στην εύρεση της βέλτιστης λύσης, αλλά γενικά στην εύρεση μίας λύσης καθώς δεν γνωρίζει την h(k) όπως ο A*.

O A* είναι ένας αλγόριθμος που εστιάζει στο να βρεί το βέλτιστο μονοπάτι. Ωστόσο, δεν γνωρίζει στην προκειμένη περίπτωση, εκ των προτέρων ότι θα βρει την βέλτιστη λύση, καθώς δεν ισχύει ότι για κάθε κατάσταση η h(k) θα είναι μικρότερη ή ίση με την πραγματική απόσταση της k από την τελική κατάσταση.

Άσκηση 2



1)
Minimax
Βήματα επίλυσης με αλγόριθμο Minimax

Βήμα 1: Εφαρμόζουμε τη συνάρτηση αξιολόγησης σε όλους τους κόμβους φύλλα του δένδρου.
Βήμα 2: Επαβαλαμβάνουμε έως ότου η ρίζα του δένδρου αποκτήσει τιμή
Βήμα 3: Καθώς ξεκινάμε από τα φύλλα του δέντρου και κινούμαστε προς τη ρίζα του, μεταφέρουμε τις τιμές προς τους ενδιάμεσους κόμβους του δέντρου.

- **Βήμα 3.α**: Η τιμή κάθε κόμβου Max είναι η μέγιστη (maximum) των τιμών των κόμβων-παιδιών του.
- **Βήμα 3.β**: Η τιμή κάθε κόμβου Min είναι η ελάχιστη (minimum) των τιμών των κόμβων-παιδιών του.

Βήμα 4:Καλύτερη κίνηση είναι η κίνηση που οδηγεί στον κόμβο που έδωσε την πιο συμφέρουσα στη ρίζα τιμή (μέγιστη για το Max, ελάχιστη για το Min).

Ακολουθούμε τον αλγόριθμο και έχοντας 4 επίπεδα άδεια, που καλούμαι να τα συμπληρώσω, αυτά αποτελούν τα εξής:

Ρίζα/Επίπεδο 0 : MAX	Επίπεδο 1 : MIN	Επίπεδο 2 : MAX	Επίπεδο 3 : MIN

Διαδοχικά από τα φύλλα, ανεβαίνουμε προς την ρίζα και συμπληρώνουμε ανά επίπεδο.

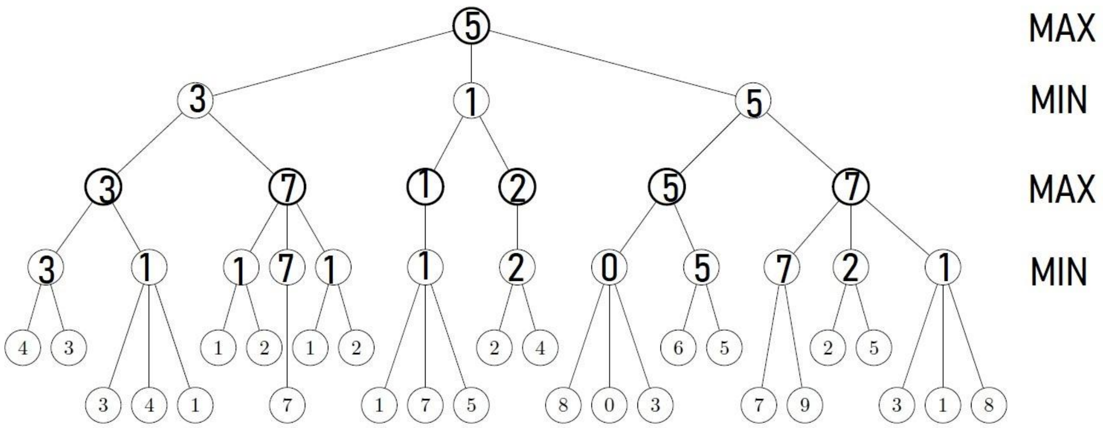
Επίπεδο 3
1ος κόμβος) min(4,3) = 3
2ος κόμβος) min(3,4,1) = 1
3ος κόμβος) min(1,2) = 1
4ος κόμβος) min(7) = 7
5ος κόμβος) min(1,2) = 1
6ος κόμβος) min(1,7,5) = 1
7ος κόμβος) min(2,4) = 2
8ος κόμβος) min(8,0,3) = 0
9ος κόμβος) min(6,5) = 5
10ος κόμβος) min(7,9) = 7
11ος κόμβος) min(2,5) = 2
12ος κόμβος) min(3,1,8) = 1

Επίπεδο 2
1ος κόμβος) max(3,1) = 3
2ος κόμβος) max(1,7,1) = 7
3ος κόμβος) max(1) = 1
4ος κόμβος) max(2) = 2
5ος κόμβος) max(0,5) = 5
6ος κόμβος) max(7,2,1) = 7

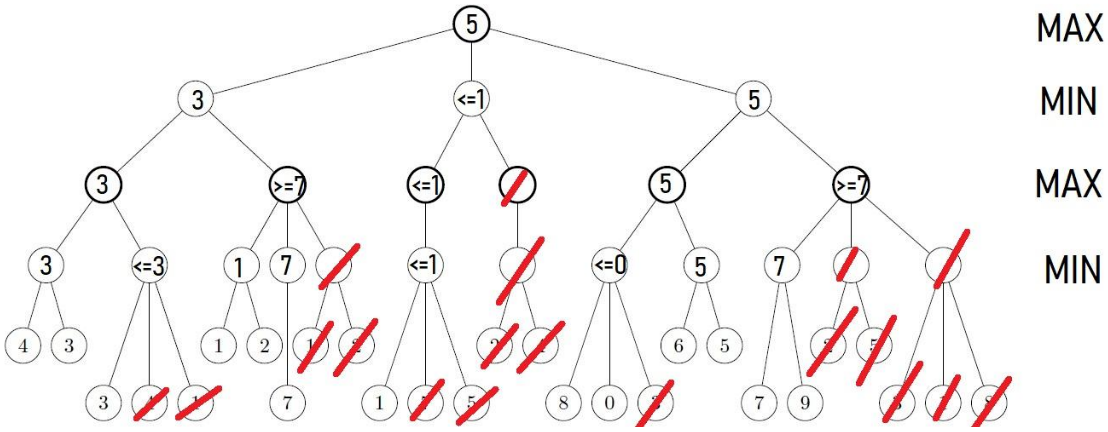
Επίπεδο 1
1ος κόμβος) min(3,7) = 3
2ος κόμβος) min(1,2) = 1
3ος κόμβος) min(5,7) = 5

Επίπεδο 0
1ος κόμβος) max(3,1,5) = 5

Άρα προκύπτει το παρακάτω



2)



ΣΕΙΡΑ ΕΠΙΣΚΕΨΗΣ ΚΟΜΒΩΝ:

1,2,5,11,23,24,12,25,6,13,28,29,14,30,3,7,16,33,4,9,18,38,39,19,41,42,10,20,43,44