ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2025 **ΣΕΙΡΑ ΑΣΚΗΣΕΩΝ 1**

Αργυρώ Τσίπη

031 19950

<u>Άσκηση 1</u>
Θα τρέξουμε και τους τρεις αλγόριμους στο πινακάκι παρακάτω.

Βήμα	Τρέχουσα	Κλειστό	Ανοιχτό	Μέτωπο Αναζήτησης	Μονοπάτι
Hill Climbing					
0	s	{s}	{a,b,c}	{a, 2} s->a {b, 5} s->b {c,6} s->c	s
1	a	{s,a}	{d,i}	{d,5} s->a->d {i,4} s->a->i	s->a
2	i	{s,a,i}	{j}	{j,3} s->a->i->j	s->a->i
3	j	{s,a,i,j}	{g}	{g,0} s->a->i->j->g	s->a->i->j
4	g	{s,a,i,j,g}	{}	{}	s->a->i->j->g
Best-First Search					
0	s	{s}	{a,b,c}	{a, 2} s->a {b, 5} s->b {c,6} s->c	s
1	a	{s,a}	{i,d,b,c}	{i,4} s->a->i {d,5} s->a->d {b, 5} s->b {c,6} s->c	s->a
2	i	{s,a,i}	{j,d,b,c}	{j,3} s->a->i->j {d,5} s->a->d {b, 5} s->b {c,6} s->c	s->a->i
3	j	{s,a,i,j}	{g,d,b,c}	{g,0} s->a->i->j- >g {d,5} s->a->d {b, 5} s->b {c,6} s->c	s->a->i->j
4	g	{s,a,i,j,g}	{d,b,c}	{}	s->a->i->j->g
A*					
0	s	{s}	{a,b,c}	{a, 2+1= 3 } s->a {b, 5+2 = 7} s->b {c, 6+1 = 7} s->c	s

Βήμα	Τρέχουσα	Κλειστό	Ανοιχτό	Μέτωπο Αναζήτησης	Μονοπάτι
1	a	{s,a}	{i,d,b,c}	{i, 4+6+1=11} s->a->i {d, 5+2+1=8} s->a->d {b, 5+2 = 7} s->b {c, 6+1 = 7} s->c	s->a
2	b	{s,a,b}	{i,d,e,k,c}	{i, 4+6+1=11} s->a->i {d, 5+2+1=8} s->a->d {e, 2+2+3 = 7} s->b->e {k, 2+2+1 = 5} s->b->k {c, 6+1 = 7} s->c	s->b
3	k	{s,a,b,k}	{i,d,e, g,j,h, c}	{i, 4+6+1=11} s->a->i {d, 5+2+1=8} s->a->d {e, 2+2+3 = 7} s->b->e {g, 0+9+1+2 = 12} s->b->k->g {j, 3+7+1+2=13} s->b->k->j {h, 7+1+1+2 = 11} s->b->k->h {c, 6+1 = 7} s->c	s->b->k

Βήμα	Τρέχουσα	Κλειστό	Ανοιχτό	Μέτωπο Αναζήτησης	Μονοπάτι
4	c	{s,a,b,k,c}	{i,d,e,g,j,h, k,i}	{i, 4+6+1=11} s->a->i {d, 5+2=7} s->c->d {e, 2+2+3 = 7} s->b->e {g, 0+9+1+2 = 12} s->b->k->g {j, 3+7+1+2= 13} s->b->k->j {h, 7+1+1+2 = 11} s->b->k->h {k, 2+1+1 = 4} s->c->k {i, 4+3+1=8} s->c->i	s->c
5	k	{s,a,b,c,k}	{i,d,e,g,j,h}	{i, 4+6+1=11} s->a->i {d, 5+2+1=8} s->a->d {e, 2+2+3 = 7} s->b->e {g, 0+9+1+1 = 11} s->c->k->g {j, 3+7+1+1=12} s->c->k->j {h, 7+1+1+1 = 10} s->c->k->h {i, 4+3+1=8} s->c->i	s->c->k

Βήμα	Τρέχουσα	Κλειστό	Ανοιχτό	Μέτωπο Αναζήτησης	Μονοπάτι
6	d	{s,a,b,k,c,d}	{e,g,j,h,i}	{e, 2+2+3 = 7} s->b->e {g,0+9+1+1=11 } s->c->k->g {j, 3+7+1+1=12} s->c->k->j {h,7+1+1+1=10 } s->c->k->h {i, 4+3+1=8} s->c->i	s->a->d
7	e	{s,a,b,k,c,d,e}	{i,g,j,h}	{g, 0+6+3+2 = 11} s->b->e->g {g,0+9+1+1=11 } s->c->k->g {j, 3+7+1+1=12} s->c->k->j {h,7+1+1+1=10 } s->c->k->h {i, 4+3+1=8} s->c->i	s->b->e
8	i	{s,a,b,k,c,d,e,i}	{g,j,h,j}	{g, 0+6+3+2 = 11} s->b->e->g {j,3+7+1+2=13 } s->b->k->j {h,7+1+1+2=11 } s->b->k->h {g,0+9+1+1=11 } s->c->k->g {h,7+1+1+1=10 } s->c->k->h {j, 3+3+3+1=10} s->c->i->j	s->c->i

Βήμα	Τρέχουσα	Κλειστό	Ανοιχτό	Μέτωπο Αναζήτησης	Μονοπάτι
9	j	{s,a,b,k,c,d,e,i,j}	{g,h}	{g, 0+6+3+2 = 11} s->b->e->g {g,0+9+1+1=11} s->c->k->g {j, 3+7+1+1=12} s->c->k->j {h,7+1+1+1=10} s->c->k->h	s->c->i->j->g
10	h	{s,a,b,k,c,d,e,i,j,h}	{g}	{i, 4+3+2+1=10} s->a->d->i {g, 0+6+3+2 = 11} s->b->e->g {g,0+9+1+1=11} s->c->k->g {j, 3+7+1+1=12} s->c->k->j {i, 4+3+1+1+1=10} s->c->k->h->i	s->c->k->h
11	i	{s,a,b,k,c,d,e,i,j,h,g}	{}	{g, 0+6+3+2 = 11} s->b->e->g {g,0+9+1+1=11} s->c->k->g	s->c->k->h->i
	answer: 2 solutions Paths: {s,b,e,g} {s,c,k,g} total cost = 11 f(n) = g(n) + c(n)				

Οι σωστές λύσεις είναι {s,b,e,g} και {s,c,k,g} με κόστος 11.

	Βρήκε βέλτιστη λύση;	**Μπορούμε να ξέρουμε από πριν εάν θα βρει βέλτιστη λύση;**
Hill Climbing	οχι , s->a->i->j->g	όχι
Best First Search	οχι , s->a->i->j->g	όχι
A*	ναι, {s,b,e,g} και {s,c,k,g}	ναι, ελέγχοντας εάν η h(n) συνεπής

Για τον A*:
O A* βρίσκει τη βέλτιστη λύση εάν η ευριστική συνάρτηση h(n) είναι συνεπής για κάθε κόμβο n.
Δηλαδή, για κάθε ακμή (n,n') πρέπει να ισχύει: h(n) <= c(n,n') + h(n')

Θα ελέγξω τώρα εάν ισχύει αυτό στο γράφημά μας:

	s	**a**	**b**	**c**	**d**	**e**	**g**	**h**	**i**	**j**	**k**
s		1	2	1							
a					2				6		
b				2		3					1
c									3		1
d									3		
e							6				
g											
h									3	7	
i										3	
j							3				
k							9				

Για την ακμή (s,a) : 10 <= 1 + 2 δεν ισχύει, άρα δεν είναι συνεπής η ευριστική.

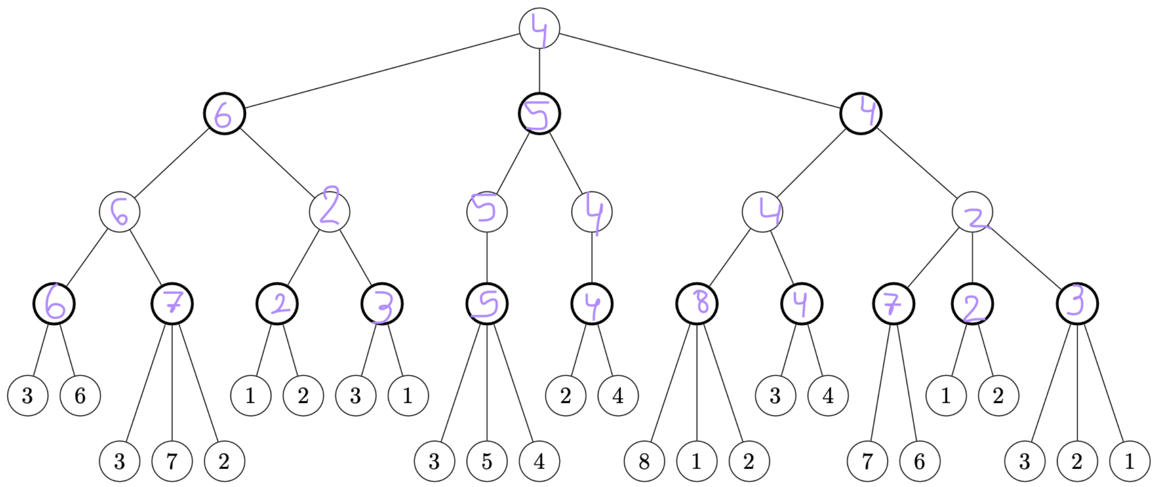
Επομένως, όταν θα τρέξουμε τον A*, και πέσουμε σε κόμβους που έχουμε επισκεφτεί, θα πρέπει να τους επισκεφτούμε ξανά. Εάν ήταν συνεπής η ευριστική τότε δε θα χρειαζόταν να τους επισκεφτούμε ξανά, θα αρκούσε μία φορά και θα τους βάζαμε στο κλειστό σύνολο και θα τελειώναμε. Μπορεί ακόμα να βρει το optimal path αλλά θα κάνει παραπάνω βήματα και επαναλήψεις.

	Βρίσκει βέλτιστη λύση;	**χρησιμοποιεί κόστος g(n);**	**χρησιμοποιεί ευριστική h(n);**	**μπορεί να κολλήσει;**
Hill Climbing	όχι πάντα (greedy)	όχι	ναι	ναι
Best First Search	όχι πάντα (greedy)	όχι	ναι	ναι
A*	ναι, εαν h(n) συνεπής + admissable	ναι	ναι	ναι

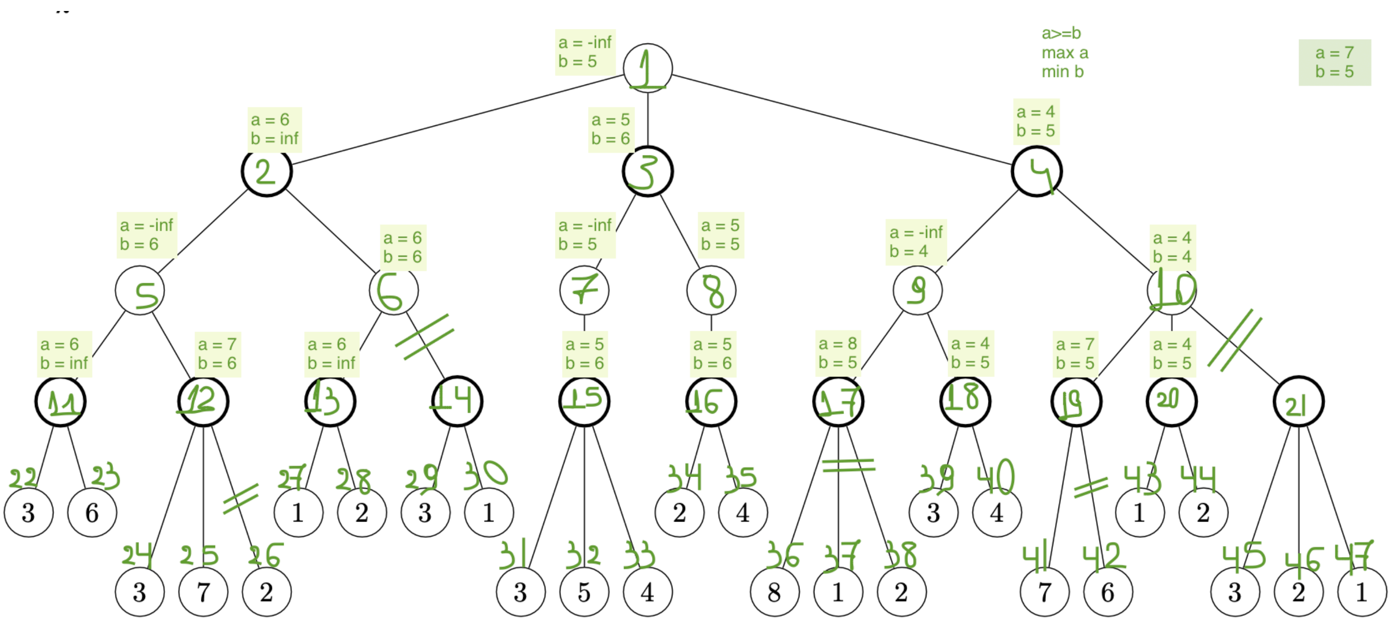
Προτεινόμενη ευρετική ώστε για κάθε ακμή (n,n') να ισχύει: h(n) <= c(n,n') + h(n'):
h(s) = 0 , h(a) = 1, h(b) = 1 , h(c) = 2 , h(e) = 2 , h(d) = 3 , h(k) = 3, h(h) = 4, h(i) = 5, h(j) = 6, h(g) = inf
Έτσι ο A* δε θα χρειάζεται να επισκέπτεται κόμβους που έχει ήδη εξερευνήσει ώστε να βρει τη σωστή λύση και άρα δε θα κάνει τόσα πολλά βήματα.

Άσκηση 2
Βήμα 1:
Αν ο A* φτάσει στον στόχο και τερματίσει αμέσως, χωρίς να συνεχίσει σε άλλους κόμβους που μπορεί να έχουν χαμηλότερο συνολικό κόστος, αυτό σημαίνει, ότι όταν έφτασε στον στόχο, δεν υπήρχαν άλλοι κόμβοι με μικρότερο f(n) που να περιμένουν να επεκταθούν. Αυτό συμβαίνει γιατί ο A* επιλέγει πάντα τους κόμβους με το μικρότερο f(n) και είναι βέβαιο ότι όταν φτάσει στον στόχο, δεν υπάρχουν άλλοι κόμβοι που να έχουν μικρότερο κόστος f(n) από τη διαδρομή που ακολουθήθηκε μέχρι τον στόχο.
Βήμα 2 (αντίστροφα):
Ας υποθέσουμε ότι ο A* τερματίζει και βρίσκει μια λύση που δεν είναι βέλτιστη. Αυτό σημαίνει ότι υπάρχει κάποιο άλλο μονοπάτι με μικρότερο συνολικό κόστος που ο αλγόριθμος δεν ανακάλυψε. Αν ο αλγόριθμος είχε επεκτείνει οποιονδήποτε άλλο κόμβο με μικρότερο f(n), θα είχε επιλέξει πρώτα αυτόν τον κόμβο, καθώς οι κόμβοι επεκτείνονται με βάση την τιμή του f(n). Εφόσον δεν επέλεξε άλλο μονοπάτι, σημαίνει ότι όλοι οι άλλοι κόμβοι είχαν υψηλότερη τιμή του f(n) και επομένως το μονοπάτι που ακολούθησε ο A* ήταν το βελτιστο.

Άσκηση 3
1.



2.
Συνθήκη a>=b, max a, min b.
με πράσινο έχω αριθμήσει τους κόμβους.



initialization:
a=-inf, b=inf για τους κόμβους 1, 2, 5, 11.

algorithm: ποιούς κόμβους επισκέπτεται μετά την αρχικοποίηση (έχω καταγράψει τους κόμβους κάθε φορά που τους επισκεπτόμαστε σε αυτήν την περίπτωση):
22, 11, 23, 11, 5, 12, 24, 12, 25, 12, 5, 2, 6, 13, 27, 13, 28, 13, 6, 2, 1, 3, 7, 15, 31, 15, 32, 15, 33, 15, 7, 3 , 8, 16, 34, 16, 35, 16, 8, 3, 1, 4 ,9, 17, 36, 17, 9, 18, 39, 18, 40, 18, 9, 4, 10, 19, 41,19, 10, 20, 43, 20, 44, 20, 10.

algorithm (καταγράφοντας κάθε κόμβο μόνο την πρώτη φορά που θα εισέλθει σε αυτόν προερχόμενος από τον πρόγονό του, εάν έχω καταλάβει καλά):
22, 23, 12, 24, 25, 6, 13, 27, 28, 3, 7, 15, 31, 32, 33 , 8, 16, 34, 35, 4 , 9, 17, 36, 18, 39, 40, 10, 19, 41, 20, 43, 44.

Δε θα εξεταστούν οι κόμβοι:
26, 14, 29, 30, 37, 38, 42, 21, 45, 46, 47.
Επειδή ισχύει εκεί α>=β.