

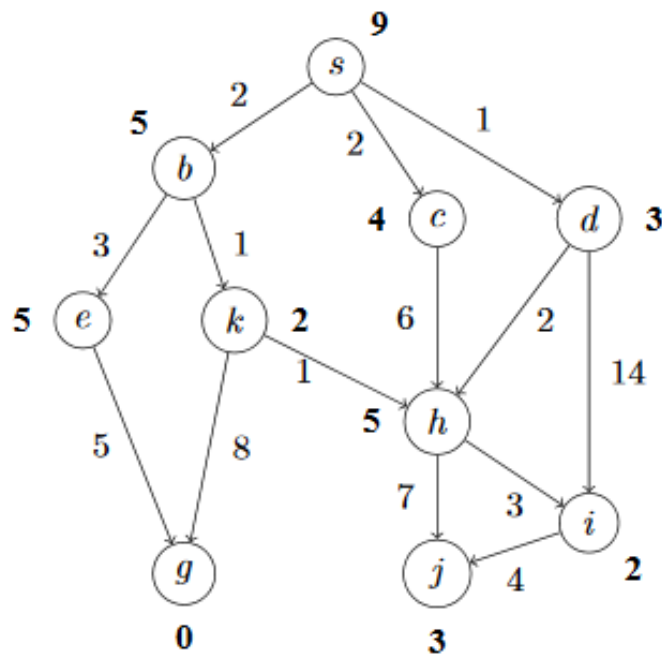
# Τεχνητή Νοημοσύνη

## 1η γραπτή σειρά ασκήσεων

Καλογερόπουλος Ιωάννης

A.M.: 03116117

### Άσκηση 1



1.

Για τον παραπάνω χώρο αναζήτησης, όπου  $s$  είναι η αρχική και  $g$  η τελική κατάσταση, εκτελούμε τους αλγορίθμους Hill Climb, Best First,  $A^*$ .

### Αλγόριθμος Αναρρίχησης λόφου (Hill Climb)

Ο αλγόριθμος Hill Climb βρίσκει κάθε φορά το παιδί της τρέχουσας κατάστασης με την ελάχιστη υπολογιζόμενη υπόλοιπη απόσταση από τον τελικό στόχο και αν αυτή είναι μικρότερη από την αντίστοιχη της τρέχουσας κατάστασης τότε το παιδί αυτό εισέρχεται στο μέτωπο αναζήτησης και γίνεται η νέα τρέχουσα κατάσταση. Στο μέτωπο αναζήτησης υπάρχει ένα παιδί κάθε φορά, το οποίο αποτελεί και την τρέχουσα κατάσταση.

Εκτελώντας τον αλγόριθμο Hill Climb, προκύπτει ο παρακάτω πίνακας:

Μέτωπο Αναζήτησης	Κλειστό σύνολο	Τρέχουσα Κατάσταση	Παιδιά τρέχουσας κατάστασης
$(s, 9)^s$	$\emptyset$	s	b(5), c(4), d(3)
$(d, 3)^{s, d}$	s	d	h(5), i(2)
$(i, 2)^{s, d, i}$	s, d	i	j(3)

Όπως, προκύπτει ο αλγόριθμος δεν καταλήγει στον τελικό στόχο αλλά σε ένα τοπικό ελάχιστο. Ο αλγόριθμος δε θα επεκταθεί και στο παιδί j του i, γιατί αυτό έχει μεγαλύτερη υπολογιζόμενη απόσταση από τον στόχο.

Τελικά, ο αλγόριθμος Hill Climb θα ανακοινώσει αποτυχία, αφού δε θα καταλήξει στην τελική κατάσταση g.

### Αλγόριθμος Best First

Ο αλγόριθμος Best First εισάγει κάθε φορά στο μέτωπο αναζήτησης τα παιδιά του κόμβου που επεκτείνει και στη συνέχεια ταξινομεί όλη τη λίστα σε αύξουσα σειρά, με βάση την ευριστική τιμή για την απόσταση από τον κόμβο στόχο. Έτσι, στην αρχή της λίστας καταλήγει ο “καλύτερος” κόμβος, δηλαδή αυτός με τη μικρότερη τιμή εκτίμησης της απόστασης από το στόχο. Ο κόμβος αυτός γίνεται η επόμενη τρέχουσα κατάσταση και σε αυτόν επεκτείνεται ο αλγόριθμος.

Εκτελώντας τον αλγόριθμο Best First προκύπτει ο παρακάτω πίνακας:

Μέτωπο Αναζήτησης	Κλειστό σύνολο	Τρέχουσα Κατάσταση	Παιδιά τρέχουσας κατάστασης
$(s, 9)^s$	$\emptyset$	s	b(5), c(4), d(3)
$(d, 3)^{s, d}, (c, 4)^{s, c}, (b, 5)^{s, b}$	s	d	h(5), i(2)
$(i, 2)^{s, d, i}, (c, 4)^{s, c}, (b, 5)^{s, b}, (h, 5)^{s, d, h}$	s, d	i	j(3)
$(j, 3)^{s, d, i, j}, (c, 4)^{s, c}, (b, 5)^{s, b}, (h, 5)^{s, d, h}$	s, d, i	j	-
$(c, 4)^{s, c}, (b, 5)^{s, b}, (h, 5)^{s, d, h}$	s, d, i, j	c	h(5)
$(b, 5)^{s, b}, (h, 5)^{s, d, h}$	s, d, i, j, c	b	e(5), k(2)
$(k, 2)^{s, b, k}, (h, 5)^{s, d, h}, (e, 5)^{s, b, e}$	s, d, i, j, c, b	k	g(0), h(5)
$(g, 0)^{s, b, k, g}, (h, 5)^{s, d, h}, (e, 5)^{s, b, e}$	s, d, i, j, c, b, k	g	-

Η κατάσταση h εμφανίστηκε δύο φορές στο μέτωπο αναζήτησης, ωστόσο κρατήθηκε η καλύτερη διαδρομή, παρόλο που στον αλγόριθμο αυτό δεν λαμβάνεται υπόψιν το μήκος μονοπατιού. Η διαγραφή αυτής της κατάστασης δεν επηρέασε το τελικό

αποτέλεσμα.

## Αλγόριθμος A\*

Ο αλγόριθμος A\* λαμβάνει υπόψιν πέρα από την ευριστική εκτίμηση και το μήκος του μονοπατιού από την αρχική μέχρι την αντίστοιχη κατάσταση. Στο μέτωπο αναζήτησης γίνεται ταξινόμηση με βάση το άθροισμα ευριστικής τιμής και μήκος μονοπατιού.

Εκτελώντας τον αλγόριθμο A\*, προκύπτει ο παρακάτω πίνακας:  
(στο μέτωπο αναζήτησης, στο δεύτερο μέλος έχουμε το ζεύγος (G;F) που δηλώνει (απόσταση από την αρχική κατάσταση ; σύνθετη ευριστική συνάρτηση  $f=g+h$ ))

Μέτωπο Αναζήτησης	Κλειστό σύνολο	Τρέχουσα Κατάσταση	Παιδιά τρέχουσας κατάστασης
$(s, 0;9)^s$	$\emptyset$	s	b(5), c(4), d(3)
$(d, 1;4)^{s,d}, (c, 2;6)^{s,c}, (b, 2;7)^{s,b}$	s	d	h(5), i(2)
$(c, 2;6)^{s,c}, (b, 2;7)^{s,b}, (h, 3;8)^{s,d,h}, (i, 15;17)^{s,d,i}$	s, d	c	h(5)
$(b, 2;7)^{s,b}, (h, 3;8)^{s,d,h}, (i, 15;17)^{s,d,i}$	s, d, c	b	e(5), k(2)
$(k, 3;5)^{s,b,k}, (h, 3;8)^{s,d,h}, (e, 5;10)^{s,b,e}, (i, 15;17)^{s,d,i}$	s, d, c, b	k	g(0), h(5)
$(h, 3;8)^{s,d,h}, (e, 5;10)^{s,b,e}, (g, 11;11)^{s,b,k,g}, (i, 15;17)^{s,d,i}$	s, d, c, b, k	h	j(3), i(2)
$(i, 6;8)^{s,d,h,i}, (e, 5;10)^{s,b,e}, (g, 11;11)^{s,b,k,g}, (j, 10;13)^{s,d,h,j}$	s, d, c, b, k, h	i	j(3)
$(e, 5;10)^{s,b,e}, (g, 11;11)^{s,b,k,g}, (j, 10;13)^{s,d,h,j}, (j, 10;13)^{s,d,h,i,j}$	s, d, c, b, k, h, i	e	g(0)
$(g, 10;10)^{s,b,e,g}, (j, 10;13)^{s,d,h,j}, (j, 10;13)^{s,d,h,i,j}$	s, d, c, b, k, h, i, e	g	-

Στις περιπτώσεις που εμφανίστηκε ή ίδια κατάσταση δύο φορές στο μέτωπο αναζήτησης κρατήθηκε αυτή με το συντομότερο μονοπάτι από την αρχική κατάσταση, ενώ σε περίπτωση ισοδύναμου μονοπατιού και οι δύο.

2.

Το πρόβλημα έχει δύο λύσεις: **s, b, e, g** και **s, b, k, g**. Η βέλτιστη λύση είναι η **s, b, e, g** αφού αυτό το μονοπάτι έχει μήκος 10, ενώ η δεύτερη χειρότερη λύση έχει μήκος 11.

Οι λύσεις που βρίσκουν οι παραπάνω αλγόριθμοι είναι:

**Hill Climb:** **s, d, i** (καταλήγει σε τοπικό ελάχιστο, **χωρίς να φτάνει στο στόχο**)

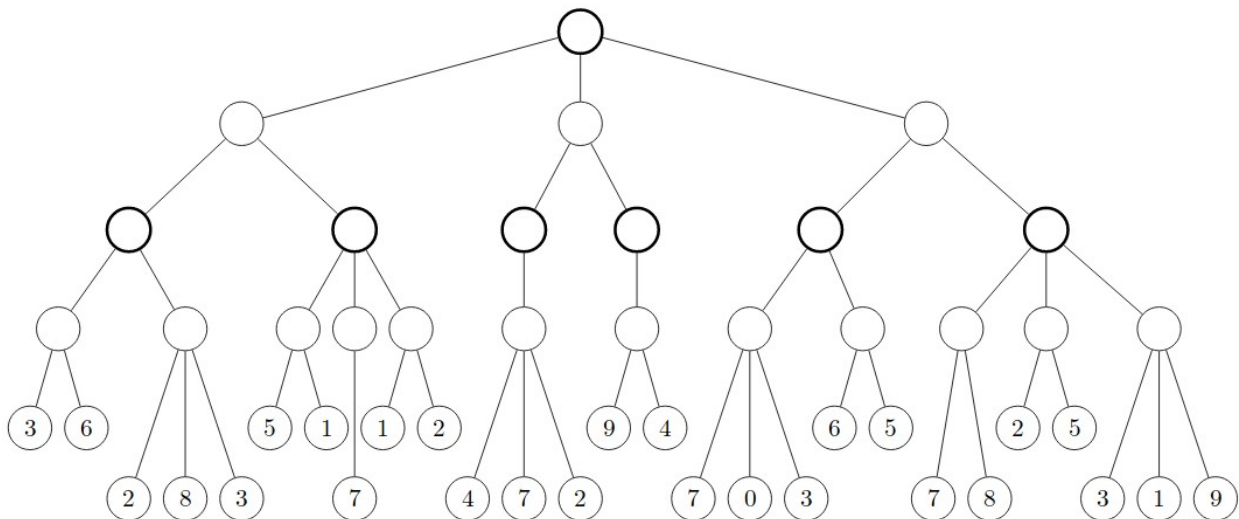
**Best First:** **s, b, k, g** (φτάνει στην τελική κατάσταση, **αλλά όχι βέλτιστα**)

**A\*:** **s, b, e, g** (φτάνει στην τελική κατάσταση, με τον **βέλτιστο τρόπο**)

Καταλήγουμε ότι ο αλγόριθμος A\* είναι ο μοναδικός που φτάνει στην τελική κατάσταση g με τη βέλτιστη λύση.

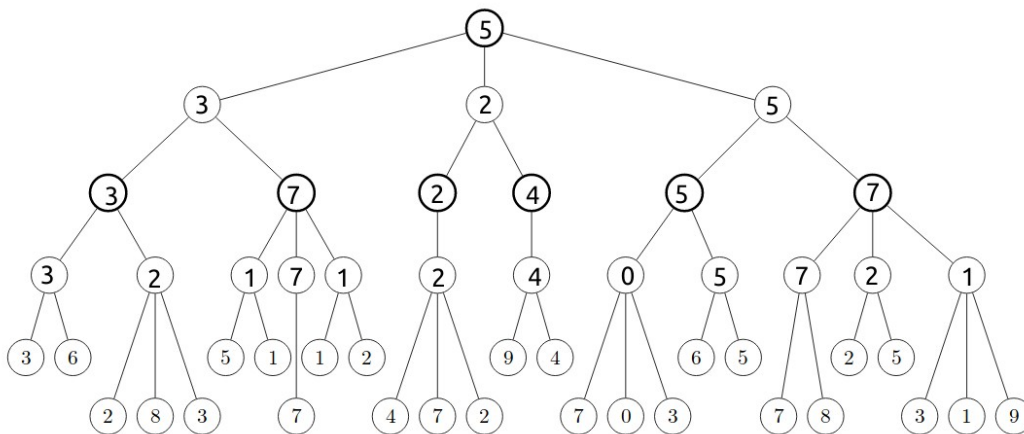
Επίσης, παρατηρούμε ότι για κάθε κατάσταση η ευριστική τιμή είναι μικρότερη ή ίση της πραγματική απόστασης από την τελική κατάσταση. Για αυτόν το λόγο ήμασταν σίγουροι ότι ο αλγόριθμος A\* θα έβρισκε τη βέλτιστη λύση.

## Άσκηση 2



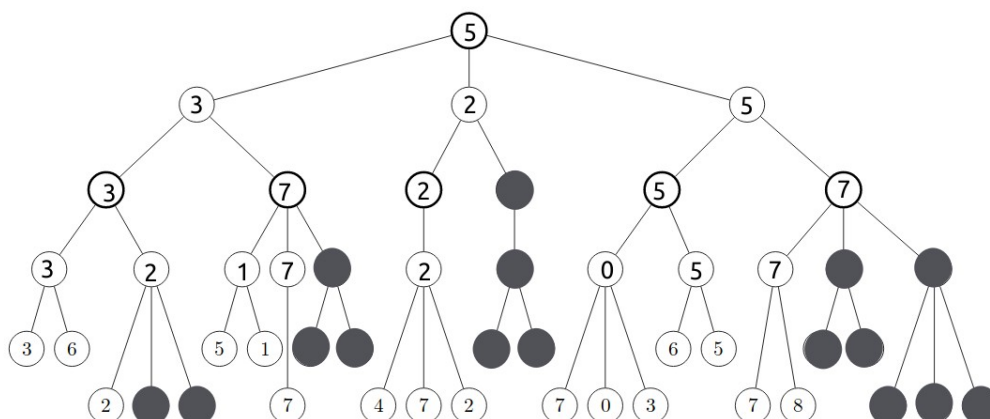
1.

Οι τιμές των κόμβων του δέντρου που έχει υπολογίσει ο αλγόριθμος Minimax φαίνονται παρακάτω:



2.

Ο αλγόριθμος Alpha-Beta (AB) θα υπολογίσει τις παρακάτω τιμές για τους κόμβους που θα επισκεφθεί. Με γκριζό χρώμα φαίνονται οι κόμβοι που **δε** θα επισκεφθεί ο αλγόριθμος AB:



Θεωρώντας αρίθμηση των κόμβων, όπως σε μία κατά πλάτος διάσχιση του δέντρου, η σειρά με την οποία θα επισκεφθεί (την πρώτη φορά) ο AB τους κόμβους είναι η εξής:

1, 2, 5, 11, 23, 24, 12, 25, 6, 13, 26, 27, 14, 30, 3, 7, 16, 33, 34, 35, 4, 9, 18, 36, 37, 38, 19, 41, 42, 10, 20, 43, 44.

Τους υπόλοιπους 16 κόμβους **δε** θα επισκεφθεί ο αλγόριθμος.

*Καλογερόπουλος Ιωάννης*

*A.M.: 03116117*