



# Επίλυση Προβλημάτων

Αποτελεί ένα από τα βασικά χαρακτηριστικά γνωρίσματα της νοημοσύνης.

Τεχνητή Νοημοσύνη = Αναπαράσταση Γνώσης + Αλγόριθμοι Αναζήτησης



# Κατηγορίες Προβλημάτων Αναζήτησης

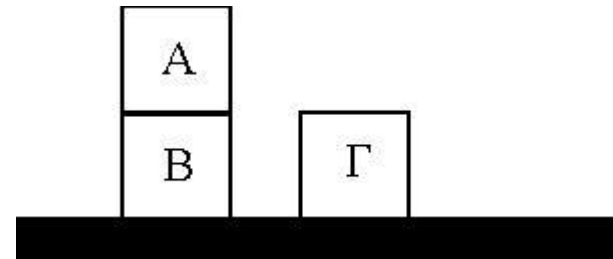
## ❖ Πραγματικά και πολύπλοκα προβλήματα (real world problems):

- ☐ σκάκι (*chess*),
- ☐ πλανόδιος πωλητής (*traveling salesperson*),
- ☐ *N*-βασίλισσες (*N-queens*),
- ☐ σάκος (*knapsack*), κλπ.
- ☐ δρομολόγηση
- ☐ χρονοπρογραμματισμός

## ❖ Απλά προβλήματα (toy problems)

- ☐ κύβοι (*blocks*),
  - ☐ *N*-puzzle,
  - ☐ τρίλιζα (*tic-tac-toe*),
  - ☐ λαβύρινθος (*maze*),
  - ☐ πύργοι του Ανόι (*Hanoi towers*),
  - ☐ κανίβαλοι και ιεραπόστολοι (*missionaries and cannibals*),
  - ☐ δοχεία (*water glass*), κλπ.
-

κύβοι (*blocks*),



*N*-puzzle,

8	3	5
4	1	7
2		6

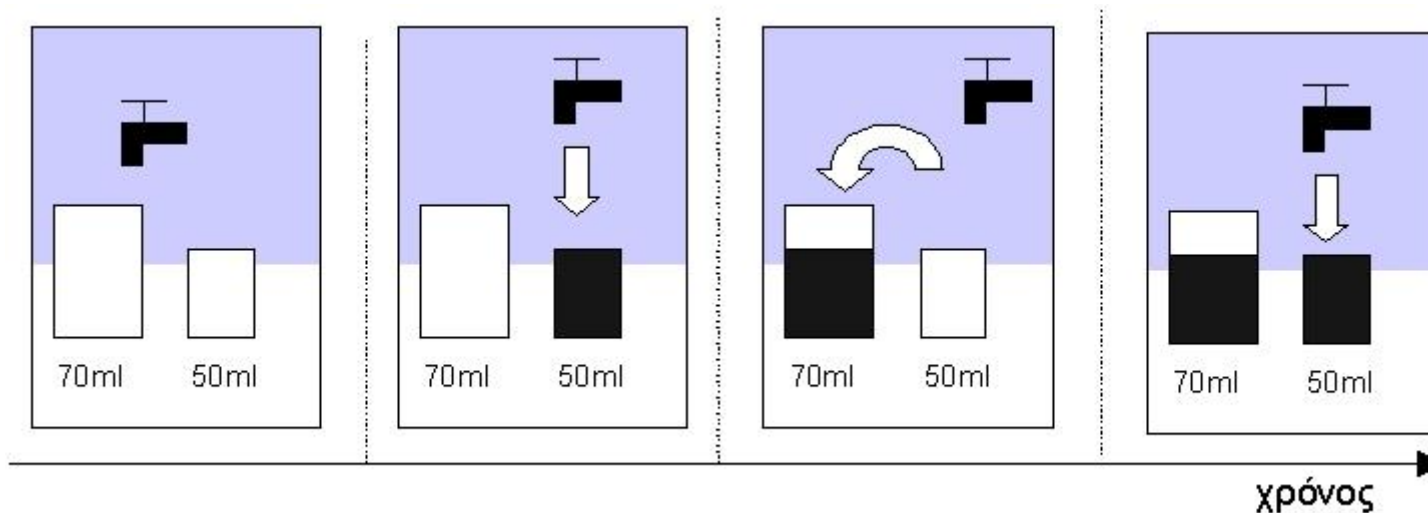
τρίλιζα (*tic-tac-toe*),

		O
O	X	
	X	

# Περιγραφή Προβλημάτων με Χώρο Καταστάσεων

## Κατάσταση προβλήματος

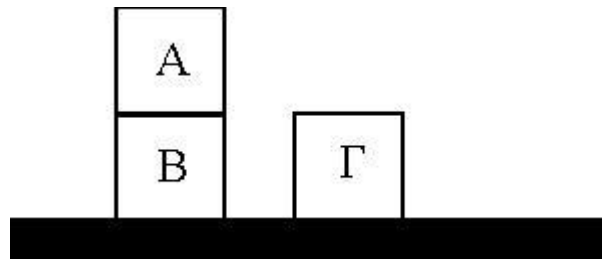
Κατάσταση ενός κόσμου είναι ένα *στιγμιότυπο (instance)* ή *φωτογραφία (snapshot)* μίας συγκεκριμένης χρονικής στιγμής της εξέλιξης του κόσμου.



Κατάσταση (*state*) ενός κόσμου είναι μία επαρκής αναπαράσταση του κόσμου σε μία δεδομένη χρονική στιγμή.



# Παράδειγμα



Αντικείμενα	Ιδιότητες	Σχέσεις
Κύβος Α	Κύβος Α είναι ελεύθερος	Κύβος Α πάνω στον κύβο Β
Κύβος Β	Κύβος Γ είναι ελεύθερος	Κύβος Β πάνω στο Τ
Κύβος Γ	Τ έχει αρκετό ελεύθερο χώρο	Κύβος Γ πάνω στο Τ
Τ είναι Τραπέζι	Κύβος Β δεν είναι ελεύθερος	



I   

K   



Αντικείμενα	Ιδιότητες	Σχέσεις
3 Ιεραπόστολοι 3 Κανίβαλοι Βάρκα Αριστερή Όχθη Δεξιά Όχθη	Βάρκα δύο ατόμων	Ιεραπόστολοι στην αριστερή όχθη Κανίβαλοι στην αριστερή όχθη Βάρκα στην αριστερή όχθη



# Τελεστές μετάβασης

- ❖ Τελεστής μετάβασης (transition operator): ενέργεια με την οποία μεταβαίνουμε από μια κατάσταση σε κάποια άλλη.

## Παράδειγμα

<b>Τελεστής:</b>
Μετέφερε δύο ιεραπόστολους από την αριστερή όχθη στη δεξιά
<b>Προϋποθέσεις:</b>
Υπάρχουν τουλάχιστον 2 ιεραπόστολοι στην αριστερή όχθη. Η βάρκα είναι στην αριστερή όχθη. Ο αριθμός των ιεραποστόλων που θα μείνουν στην αριστερή όχθη να μην είναι μικρότερος από τον αριθμό των κανιβάλων ή να μην υπάρχει άλλος ιεραπόστολος στην αριστερή όχθη.
<b>Αποτελέσματα:</b>
Ο αριθμός των ιεραποστόλων στην αριστερή όχθη μειώνεται κατά 2. Ο αριθμός των ιεραποστόλων στην δεξιά όχθη αυξάνεται κατά 2. Η βάρκα είναι πλέον δεξιά και όχι αριστερά



# Χώρος Καταστάσεων

- ❖ Χώρος καταστάσεων (state space ή domain space) ενός προβλήματος ονομάζεται το σύνολο όλων των έγκυρων καταστάσεων.

## Αρχικές και Τελικές καταστάσεις

- ❖ Η αρχική (*initial state*) και τελική (*final* ή *goal state*) κατάσταση εκφράζουν το δεδομένο και το ζητούμενο αντίστοιχα.
- ❖ Οι τελικές καταστάσεις μπορεί να είναι περισσότερες από μία.

8	3	5
4	1	7
2		6

Αρχική Κατάσταση

1	2	3
4	5	6
7	8	

Τελική Κατάσταση





# Ορισμός προβλήματος

❖ Ένα πρόβλημα (*Problem*) ορίζεται ως η τετράδα  $P = (I, G, T, S)$  όπου:

- ❑  $S$  είναι ο χώρος καταστάσεων.
- ❑  $T$  είναι το σύνολο των τελεστών μετάβασης,  $T: S \leftrightarrow S$ .
  - ❑ Συνήθως δίνονται ως κανόνες οι οποίοι καθορίζουν τις επιτρεπτές ενέργειες σε κάθε κατάσταση.
  - ❑ Εφαρμόζοντας μια επιτρεπτή ενέργεια σε μια κατάσταση μεταβαίνουμε σε μια νέα κατάσταση που θεωρείται παιδί της προηγούμενης.
  - ❑ Οι ενέργειες μπορεί να έχουν κόστος (υποχρεωτικά μεγαλύτερο του μηδενός). Εάν δεν δίνεται το κόστος ενεργειών θεωρούμε ότι όλες οι ενέργειες έχουν κόστος 1.
- ❑  $I$  είναι η **αρχική κατάσταση** (A.K),  $I \in S$
- ❑  $G$  είναι το **σύνολο των τελικών καταστάσεων** (T.K),  $G \subseteq S$



# Λύση προβλήματος

Λύση (*Solution*) σε ένα πρόβλημα  $(I, G, T, S)$ , είναι μία ακολουθία από τελεστές μετάβασης  $t_1, t_2, \dots, t_n$   $\in T$  με την ιδιότητα  $g = t_n(\dots(t_2(t_1(I)))\dots)$ , όπου  $g \in G$

## Παράδειγμα

Μετέφερε 1 ιεραπόστολο και 1 κανίβαλο από την αριστερή στη δεξιά όχθη  
Μετέφερε 1 ιεραπόστολο από τη δεξιά στην αριστερή όχθη  
Μετέφερε 2 κανίβαλους από την αριστερή στη δεξιά όχθη  
Μετέφερε 1 κανίβαλο από τη δεξιά στην αριστερή όχθη  
Μετέφερε 2 ιεραπόστολους από την αριστερή στη δεξιά όχθη  
Μετέφερε 1 ιεραπόστολο και 1 κανίβαλο από τη δεξιά στην αριστερή όχθη  
Μετέφερε 2 ιεραπόστολους από την αριστερή στη δεξιά όχθη  
Μετέφερε 1 κανίβαλο από τη δεξιά στην αριστερή όχθη  
Μετέφερε 2 κανίβαλους από την αριστερή στη δεξιά όχθη  
Μετέφερε 1 ιεραπόστολο από τη δεξιά στην αριστερή όχθη  
Μετέφερε 1 ιεραπόστολο και 1 κανίβαλο από την αριστερή στη δεξιά όχθη



# Αλγόριθμοι Αναζήτησης

## Τυφλοί (blind search)

Όνομα Αλγορίθμου	Συντομογραφία	Ελληνική Ορολογία
Depth-First Search	DFS	Αναζήτηση Πρώτα σε Βάθος
Breadth-First Search	BFS	Αναζήτηση Πρώτα σε Πλάτος
Uniform Cost Search	UCS	Αναζήτηση Ομοιόμορφου Κόστους
Iterative Deepening	ID	Επαναληπτική Εμβάθυνση
Bi-directional Search	BiS	Αναζήτηση Διπλής Κατεύθυνσης

## Ευρετικοί (heuristic search)

Greedy Search	GS	Άπληστη Αναζήτηση
A* (A-star)	A*	A* (Άλφα Άστρο)



## Αναζήτηση ως διακριτή βελτιστοποίηση

Local Search	LS	Τοπική Αναζήτηση
Tabu Search	TS	Αναζήτηση με Απαγορεύσεις
Simulated Annealing	SA	Προσομοιούμενη Ανόπτηση

## Προβλήματα Ικανοποίησης Περιορισμών (Constraint Satisfaction Problems)

### Παίγνια 2 παιχτών

Minimax	Minimax	Αναζήτηση Μεγίστου-Ελαχίστου
Alpha-Beta Pruning	AB	Κλάδεμα Άλφα-Βήτα

---



# Χώρος Αναζήτησης

Δοθέντος ενός προβλήματος  $(I, G, T, S)$ , *χώρος αναζήτησης* (*search space*)  $SP$  είναι το σύνολο όλων των καταστάσεων που είναι προσβάσιμες από την αρχική κατάσταση.

Μία κατάσταση  $s$  ονομάζεται *προσβάσιμη* (*accessible*) αν υπάρχει μια ακολουθία τελεστών μετάβασης  $t_1, t_2, \dots, t_k \in T$  τέτοια ώστε  $s = t_k(\dots(t_2(t_1(I)))\dots)$ .

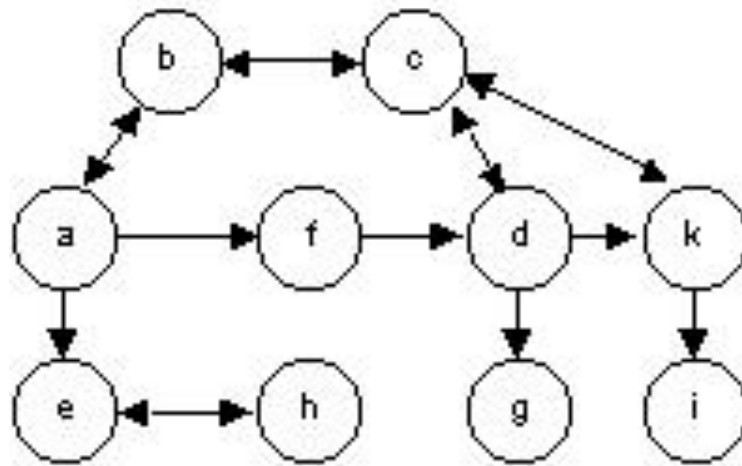
- ❖ Ο χώρος αναζήτησης είναι υποσύνολο του χώρου καταστάσεων, δηλαδή  $SP \subseteq S$ .
- ❖ Ο χώρος αναζήτησης μπορεί να αναπαρασταθεί με γράφο. Είναι πάντα εφικτό να μετατραπεί ο γράφος σε *δένδρο αναζήτησης* (*search tree*), το οποίο όμως μπορεί να έχει μονοπάτια απείρου μήκους.



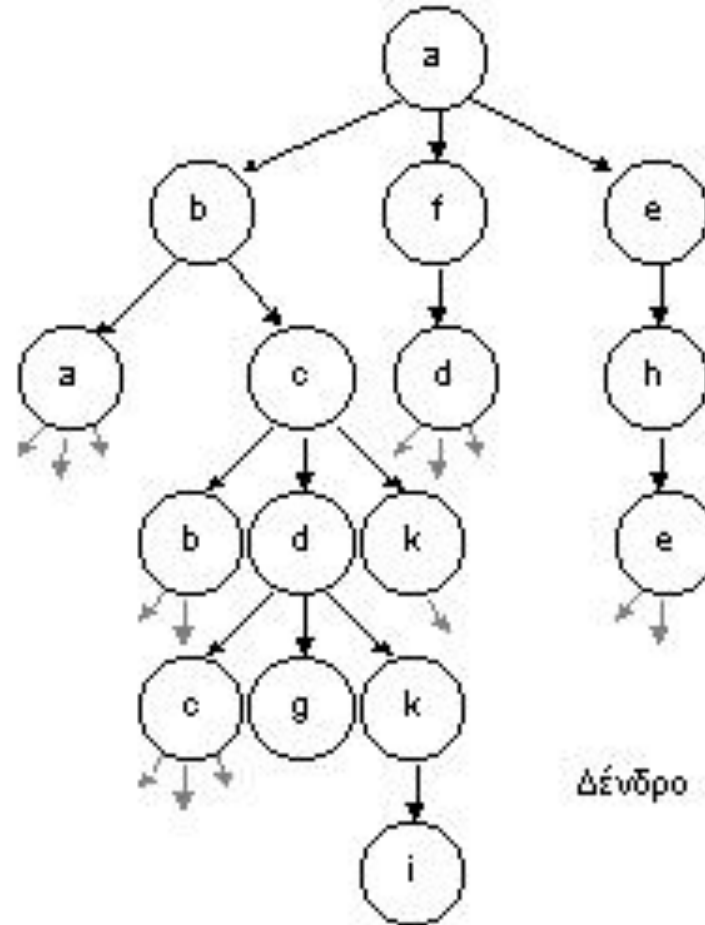
Τμήμα Δένδρου	Αναπαράσταση
Κόμβος (Node)	Κατάσταση
Ρίζα (Root)	Αρχική Κατάσταση
Φύλλο (Tip, Leaf)	Τελική Κατάσταση ή Αδιέξοδο (Dead Node), δηλαδή κατάσταση στην οποία δεν μπορεί να εφαρμοστεί κανένας τελεστής μετάβασης.
Κλαδί (Branch)	Τελεστής Μετάβασης που μετατρέπει μια κατάσταση-Γονέα (Parent State) σε μία άλλη κατάσταση-Παιδί (Child State).
Λύση (Solution)	Μονοπάτι (Path) που ενώνει την αρχική με μία τελική κατάσταση
Επέκταση (Expansion)	Η διαδικασία παραγωγής όλων των καταστάσεων-παιδιών ενός κόμβου.
Παράγοντας Διακλάδωσης (Branching Factor)	Ο αριθμός των καταστάσεων-Παιδιών που προκύπτουν από μία επέκταση. Επειδή δεν είναι σταθερός αριθμός, αναφέρεται και ως Μέσος Παράγοντας Διακλάδωσης (Average Branching Factor).

- ❖ Ο παράγοντας διακλάδωσης (branching factor) εκφράζει τον αριθμό των καταστάσεων που προκύπτουν από μία άλλη κατάσταση.

# Χώρος Αναζήτησης (Παράδειγμα)



Γράφος



Δένδρο

Το φαινόμενο της εκθετικής αύξησης του αριθμού των κόμβων του δένδρου αναζήτησης ονομάζεται συνδυαστική έκρηξη (combinatorial explosion).



# Γενικός Αλγόριθμος Αναζήτησης

Μέτωπο της αναζήτησης (search frontier) ενός αλγορίθμου είναι το σύνολο των καταστάσεων που ο αλγόριθμος έχει ήδη επισκεφτεί, αλλά δεν έχουν ακόμη επεκταθεί (φύλλα του δέντρου αναζήτησης).

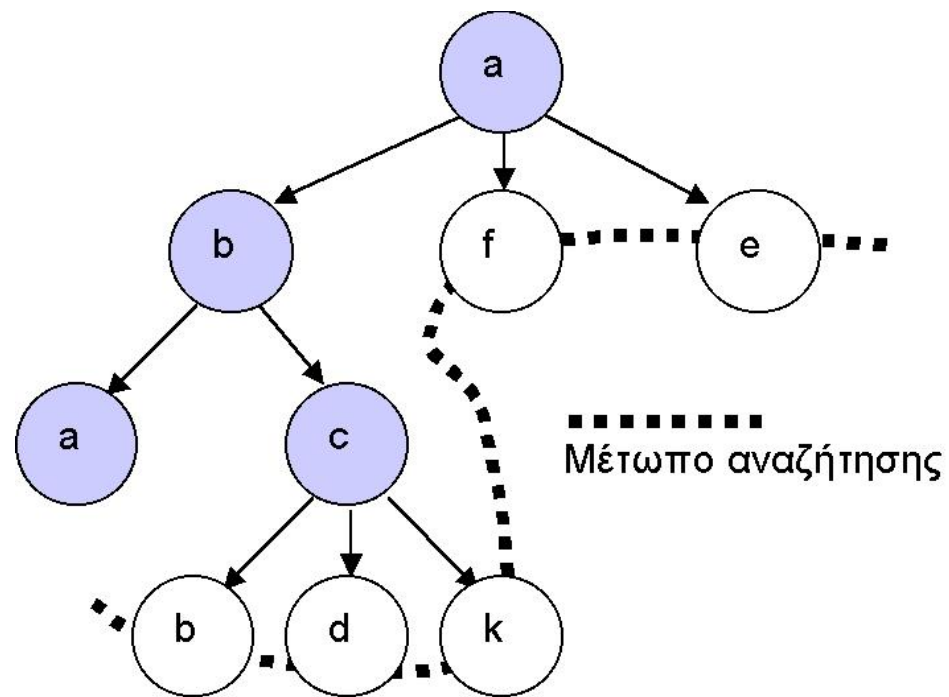
Κλειστό σύνολο (closed set) ενός αλγορίθμου αναζήτησης είναι το σύνολο όλων των καταστάσεων που έχουν ήδη επεκταθεί από τον αλγόριθμο αναζήτησης.

Με έναν απλό έλεγχο, αν η κατάσταση προς επέκταση ανήκει ήδη στο κλειστό σύνολο, αποφεύγονται οι βρόχοι (*loops*).

Το κλειστό σύνολο είναι προαιρετικό, διότι αυξάνει κατά πολύ τις απαιτήσεις σε μνήμη. Στις ασκήσεις θα θεωρούμε ότι δεν χρησιμοποιείται κλειστό σύνολο.

---





# Γενικός Αλγόριθμος Αναζήτησης

1. Βάλε την αρχική κατάσταση στο μέτωπο της αναζήτησης.
2. Αν το μέτωπο αναζήτησης είναι άδειο τότε σταμάτησε.
3. Πάρε την πρώτη σε σειρά κατάσταση του μετώπου της αναζήτησης.
4. Αν είναι η κατάσταση αυτή μέρος του κλειστού συνόλου τότε πήγαινε στο βήμα 2. (Προαιρετικό)
5. Αν είναι η κατάσταση αυτή τελική κατάσταση τότε τύπωσε τη λύση και τερμάτισε.
6. Εφάρμοσε τους τελεστές μετάβασης για να παράγεις τις καταστάσεις-παιδιά.
7. Βάλε τις νέες καταστάσεις-παιδιά στο μέτωπο της αναζήτησης (σύμφωνα με κάποιο κριτήριο).
8. Βάλε την κατάσταση-γονέα στο κλειστό σύνολο. (Προαιρετικό)
9. Πήγαινε στο βήμα 2.

Το κριτήριο με το οποίο επιλέγεται η κατάσταση του μετώπου αναζήτησης που θα επιλεγεί για επέκταση διαφοροποιεί τις μεθόδους αναζήτησης μεταξύ τους

Μια κατάσταση εξετάζεται εάν είναι Τελική Κατάσταση κατά την εξαγωγή της και όχι κατά την εισαγωγή της στο μέτωπο αναζήτησης.



# Χαρακτηριστικά Μεθόδων Αναζήτησης

- ❖ Μια μέθοδος αναζήτησης ονομάζεται **πλήρης** (complete) αν εγγυάται ότι θα βρει μία λύση για οποιαδήποτε τελική κατάσταση, αν τέτοια λύση υπάρχει. Σε αντίθετη περίπτωση, η μέθοδος ονομάζεται ατελής (incomplete).
  - ❖ Μία λύση ονομάζεται **βέλτιστη** (optimal) εάν έχει το μικρότερο κόστος μονοπατιού.
  - ❖ Μια μέθοδος αναζήτησης ονομάζεται **πλήρης και βέλτιστη** εάν εγγυάται την εύρεση λύσης και επιπλέον εγγυάται ότι η πρώτη λύση που θα βρει θα είναι και η καλύτερη δυνατή (βέλτιστη)
-



# Κριτήρια Αξιολόγησης Μεθόδων Αναζήτησης

Η επιλογή μιας μεθόδου βασίζεται στα εξής κριτήρια:

- ☐ δυνατότητα εύρεσης λύσεων εφόσον αυτές υπάρχουν (πληρότητα)
  - ☐ ποιότητα των λύσεων (βελτιστότητα)
  - ☐ πολυπλοκότητα σε χρόνο (αριθμός επεκτάσεων)
  - ☐ πολυπλοκότητα σε χώρο (μνήμη για την αποθήκευση του μετώπου αναζήτησης)
  - ☐ ευκολία υλοποίησης
-



# Αλγόριθμοι Τυφλής Αναζήτησης

Οι αλγόριθμοι τυφλής αναζήτησης (*blind search algorithms*) **δεν αξιοποιούν** πληροφορία που να επιτρέπει την αξιολόγηση των καταστάσεων του χώρου αναζήτησης σε σχέση με την απόστασή τους από τις τελικές καταστάσεις.

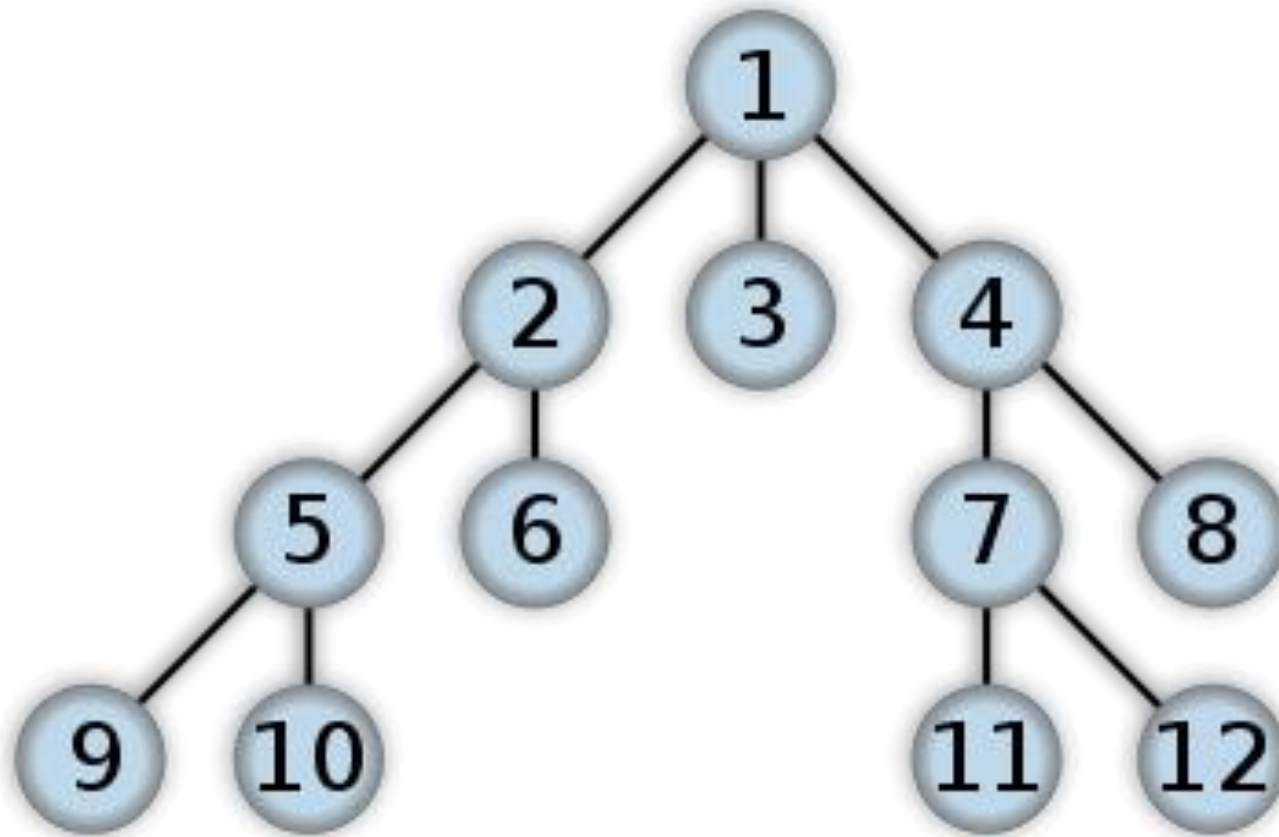
---

# Αναζήτηση Πρώτα σε Πλάτος

Ο αλγόριθμος αναζήτησης πρώτα σε πλάτος (*Breadth First Search - BFS*) εξετάζει πρώτα όλες τις καταστάσεις που βρίσκονται στο ίδιο βάθος και μετά συνεχίζει στην επέκταση καταστάσεων στο αμέσως επόμενο επίπεδο.

## Ο αλγόριθμος BFS

1. Βάλε την αρχική κατάσταση στο μέτωπο της αναζήτησης.
2. Αν το μέτωπο της αναζήτησης είναι κενό τότε σταμάτησε.
3. Βγάλε την πρώτη κατάσταση από το μέτωπο της αναζήτησης.
4. Αν είναι η κατάσταση μέλος του κλειστού συνόλου τότε πήγαινε στο βήμα 2.
5. Αν η κατάσταση είναι μία τελική τότε ανέφερε τη λύση.
6. Αν θέλεις και άλλες λύσεις πήγαινε στο βήμα 2. Αλλιώς σταμάτησε.
7. Εφάρμοσε τους τελεστές μεταφοράς για να βρεις τις καταστάσεις-παιδιά.
8. Βάλε τις καταστάσεις-παιδιά στο τέλος του μετώπου της αναζήτησης.
9. Βάλε την κατάσταση-γονέα στο κλειστό σύνολο.
10. Πήγαινε στο βήμα 2.



BFS

Οι αριθμοί δηλώνουν τη σειρά με την οποία επεκτείνονται οι κόμβοι



# Αναζήτηση Πρώτα σε Πλάτος

## Σχόλια

- ❖ Το μέτωπο της αναζήτησης είναι μια δομή ουράς (Queue FIFO, δηλαδή First In First Out).
  - ❖ Επιλογή για επέκταση της κατάστασης του μετώπου αναζήτησης με το μικρότερο βάθος
  - ❖ Πλεονεκτήματα:
    - ❑ Είναι πλήρης και βέλτιστη.
  - ❖ Μειονεκτήματα:
    - ❑ Το μέτωπο της αναζήτησης μεγαλώνει πολύ σε μέγεθος (εκθετική πολυπλοκότητα χώρου).
-



# Αναζήτηση Ομοιόμορφου Κόστους (uniform cost search)

Επιλέγεται για επέκταση ο κόμβος του μετώπου αναζήτησης με το μικρότερο  $g(n)$ , όπου  $g(n)$  είναι το κόστος του μονοπατιού από την ρίζα (Α.Κ) μέχρι τον κόμβο  $n$ .

Η μέθοδος BFS είναι ειδική περίπτωση όταν όλες οι ενέργειες έχουν το ίδιο κόστος.

Στην περίπτωση αυτή  $g(n)=\text{depth}(n)$

## Ο αλγόριθμος UCS

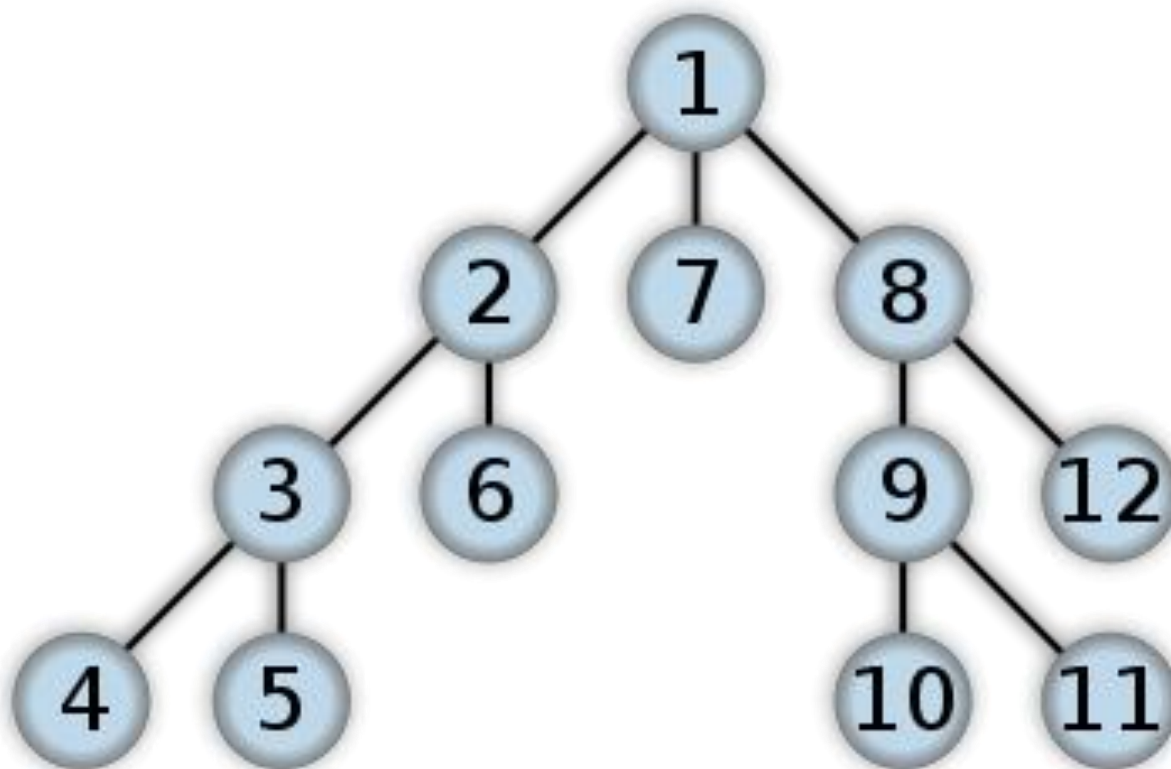
1. Βάλε την αρχική κατάσταση στο μέτωπο της αναζήτησης.
2. Αν το μέτωπο της αναζήτησης είναι κενό τότε σταμάτησε.
3. Βγάλε την πρώτη κατάσταση από το μέτωπο της αναζήτησης (κόμβος με το μικρότερο  $g(n)$ ).
4. Αν είναι η κατάσταση μέλος του κλειστού συνόλου τότε πήγαινε στο βήμα 2.
5. Αν η κατάσταση είναι μία τελική τότε ανέφερε τη λύση.
6. Αν θέλεις και άλλες λύσεις πήγαινε στο βήμα 2. Αλλιώς σταμάτησε.
7. Εφάρμοσε τους τελεστές μεταφοράς για να βρεις τις καταστάσεις-παιδιά.
8. Βάλε τις καταστάσεις-παιδιά στο μέτωπο αναζήτησης με φθίνουσα σειρά βάσει του κόστους του μονοπατιού  $g(n)$  από τη ρίζα μέχρι τον κόμβο  $n$ .
9. Βάλε την κατάσταση-γονέα στο κλειστό σύνολο.
10. Πήγαινε στο βήμα 2.

# Αναζήτηση Πρώτα σε Βάθος (DFS)

Ο αλγόριθμος πρώτα σε βάθος (*Depth-First Search - DFS*) επιλέγει προς επέκταση την κατάσταση με το μεγαλύτερο βάθος.

## Ο αλγόριθμος DFS:

1. Βάλε την αρχική κατάσταση στο μέτωπο της αναζήτησης.
2. Αν το μέτωπο της αναζήτησης είναι κενό τότε σταμάτησε.
3. Βγάλε την πρώτη κατάσταση από το μέτωπο της αναζήτησης.
4. Αν είναι η κατάσταση μέλος του κλειστού συνόλου τότε πήγαινε στο βήμα 2.
5. Αν η κατάσταση είναι μία από τις τελικές, τότε ανέφερε τη λύση.
6. Αν θέλεις και άλλες λύσεις πήγαινε στο βήμα 2. Αλλιώς σταμάτησε.
7. Εφάρμοσε τους τελεστές μετάβασης για να βρεις τις καταστάσεις-παιδιά.
8. Βάλε τις καταστάσεις-παιδιά στην αρχή του μετώπου της αναζήτησης.
9. Βάλε την κατάσταση-γονέα στο κλειστό σύνολο.
10. Πήγαινε στο βήμα 2.



DFS

Οι αριθμοί δηλώνουν τη σειρά με την οποία επεκτείνονται οι κόμβοι



## Αναζήτηση Πρώτα σε Βάθος

- ❖ Το μέτωπο της αναζήτησης είναι μια δομή στοίβας (Stack LIFO, Last In First Out)
- ❖ Η εξέταση αμέσως προηγούμενων (χρονικά) καταστάσεων ονομάζεται χρονική οπισθοδρόμηση (chronological backtracking).
- ❖ Πλεονεκτήματα:
  - ❑ Έχει μικρές απαιτήσεις σε χώρο διότι το μέτωπο της αναζήτησης δε μεγαλώνει πάρα πολύ (γραμμική πολυπλοκότητα χώρου).
- ❖ Μειονεκτήματα:
  - ❑ Δεν εγγυάται ότι η πρώτη λύση που θα βρεθεί είναι η βέλτιστη (μονοπάτι με το μικρότερο μήκος ή με μικρότερο κόστος).
  - ❑ Δεν είναι πλήρης, εκτός εάν γίνεται έλεγχος βρόχων με χρήση κλειστού συνόλου (όμως στην περίπτωση αυτή η πολυπλοκότητα χώρου γίνεται εκθετική).

## Αναζήτηση DFS με όριο βάθους (DFS(dl))

Ιδιος με τον DFS με τη διαφορά ότι υπάρχει όριο βάθους dl (depth limit). Δεν επεκτείνονται κόμβοι που βρίσκονται στο όριο βάθους dl.



# Αλγόριθμος Επαναληπτικής Εμβάθυνσης

Ο αλγόριθμος επαναληπτικής εμβάθυνσης (Iterative Deepening - *ID*) συνδυάζει με τον καλύτερο τρόπο τους DFS και BFS.

## Ο αλγόριθμος ID:

1. Όρισε το αρχικό βάθος αναζήτησης  $d1$  (συνήθως 1).
2. Εφάρμοσε τον αλγόριθμο DFS μέχρι αυτό το βάθος αναζήτησης ( $BFS(d1)$ ).
3. Αν έχεις βρει λύση σταμάτησε.
4. Αύξησε το βάθος αναζήτησης (συνήθως κατά 1) ( $d1 := d1 + 1$ ).
5. Πήγαινε στο βήμα 2.

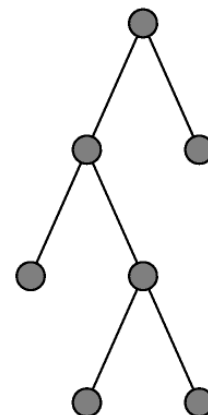
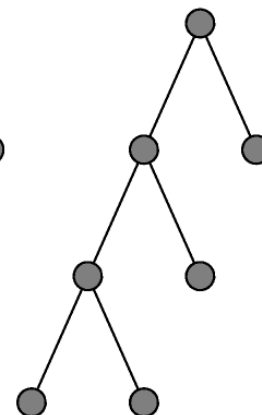
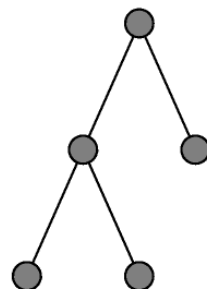
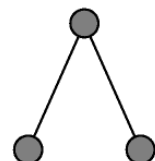
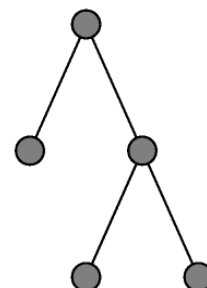
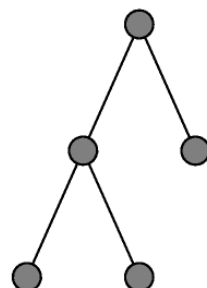
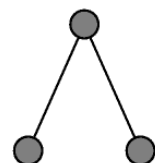
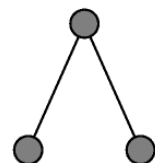


Limit = 0 ●

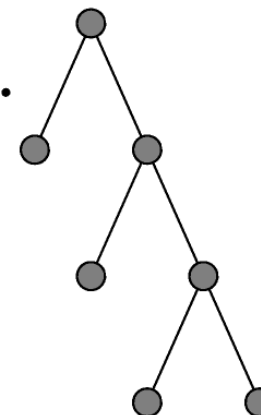
Limit = 1 ●

Limit = 2 ●

Limit = 3 ●



.....





# Αναζήτηση ID

## Σχόλια

### ❖ Μειονεκτήματα:

- ☐ Όταν αρχίζει ο DFS με διαφορετικό βάθος δε θυμάται τίποτα από την προηγούμενη αναζήτηση (μεγάλος αριθμός επεκτάσεων).

### ❖ Πλεονεκτήματα:

- ☐ Είναι πλήρης .
- ☐ Αν το βάθος αυξάνεται κατά 1 σε κάθε κύκλο και ο ID βρει λύση, τότε αυτή η λύση θα είναι η καλύτερη (βέλτιστη μέθοδος).
- ☐ Έχει γραμμική πολυπλοκότητα μνήμης (λόγω του DFS).



# Αναζήτηση Διπλής Κατεύθυνσης

Η ιδέα της αναζήτησης διπλής κατεύθυνσης (Bidirectional Search - *BiS*) πηγάζει από τη δυνατότητα του παραλληλισμού (parallelism) στα υπολογιστικά συστήματα.

- ❖ Προϋποθέσεις κάτω από τις οποίες μπορεί να εφαρμοστεί:
  - ❑ Οι τελεστές μετάβασης είναι αντιστρέψιμοι (reversible).
- ❖ Αναπτύσσονται δύο δέντρα αναζήτησης: ένα από Α.Κ προς Τ.Κ και ένα από Τ.Κ προς Α.Κ.
- ❖ Πλεονέκτημα: μειώνεται στο μισό ο χρόνος αναζήτησης
- ❖ Μειονέκτημα: είναι δύσκολη η υλοποίηση



