



Πολυτεχνείο
Κρήτης

Σχολή Μηχανικών Παραγωγής και Διοίκησης

*Εισαγωγή στην Τεχνητή
Νοημοσύνη*

Αλγόριθμοι τυφλής αναζήτησης

Τσαφαράκης Στέλιος

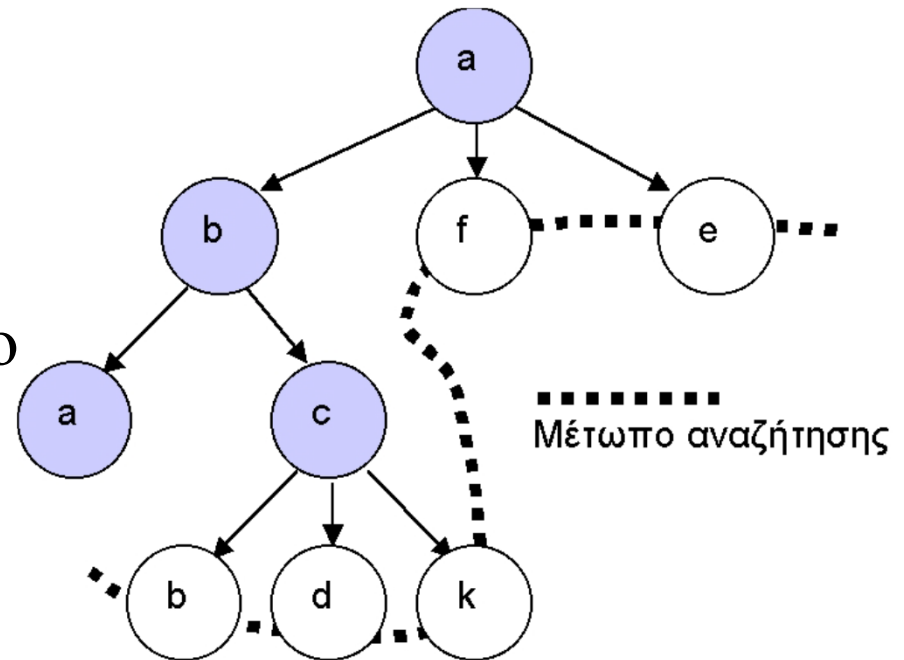
Αλγόριθμοι Τυφλής Αναζήτησης

- Οι αλγόριθμοι τυφλής αναζήτησης (*blind search algorithms*) εφαρμόζονται σε προβλήματα στα οποία δεν υπάρχει πληροφορία που να επιτρέπει αξιολόγηση των καταστάσεων.
- Στους αλγορίθμους τυφλής αναζήτησης έχει σημασία η χρονική σειρά με την οποία παράγονται οι καταστάσεις από το μηχανισμό επέκτασης.

Όνομα Αλγορίθμου	Συντομογραφία	Ελληνική Ορολογία
Depth-First Search	DFS	Αναζήτηση Πρώτα κατά Βάθος
Breadth-First Search	BFS	Αναζήτηση Πρώτα κατά Πλάτος
Iterative Deepening	ID	Επαναληπτική Εκβάθυνση
Bi-directional Search	BiS	Αναζήτηση Διπλής Κατεύθυνσης
Branch and Bound	B&B	Επέκταση και Οριοθέτηση
Beam Search	BS	Ακτινωτή Αναζήτηση

Δομές δεδομένων αλγορίθμων αναζήτησης

- *Μέτωπο της αναζήτησης* (search frontier) ενός αλγορίθμου είναι η λίστα των καταστάσεων που ο αλγόριθμος έχει ήδη επισκεφτεί, αλλά δεν έχουν ακόμη επεκταθεί.
- *Κλειστό σύνολο* (closed set) ενός αλγορίθμου αναζήτησης είναι το σύνολο όλων των καταστάσεων που έχουν ήδη επεκταθεί.
- Σε κάθε επέκταση μιας κατάστασης:
 - Εισάγεται η κατάσταση-γονέας στο κλειστό σύνολο
 - Εισάγονται οι καταστάσεις-παιδιά στο μέτωπο αναζήτησης



Χαρακτηριστικά αλγορίθμων

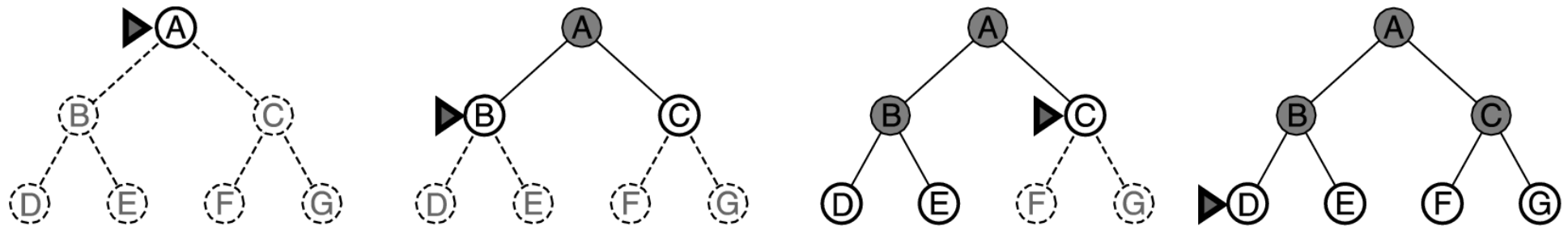
- Ένας αλγόριθμος ονομάζεται **εξαντλητικός** (*exhaustive*) όταν το σύνολο των καταστάσεων που εξετάζει για να βρει τις απαιτούμενες λύσεις είναι ίσο με το χώρο αναζήτησης
- Ένας αλγόριθμος αναζήτησης ονομάζεται **πλήρης** (*complete*) αν εγγυάται ότι θα βρει μία λύση για οποιαδήποτε τελική κατάσταση, αν τέτοια λύση υπάρχει.
- Μία λύση ονομάζεται **βέλτιστη** (*optimal*) αν οδηγεί στην καλύτερη, σύμφωνα με τη διάταξη, τελική κατάσταση.
- Όταν δεν υπάρχει διάταξη, μία λύση ονομάζεται βέλτιστη αν είναι η συντομότερη (*shortest*).
- Ένας αλγόριθμος αναζήτησης καλείται **αποδεκτός** (*admissible*) αν εγγυάται ότι θα βρει τη βέλτιστη λύση, αν μια τέτοια λύση υπάρχει.

Αναζήτηση πρώτα κατά πλάτος (Breadth-first search)

Ο αλγόριθμος αναζήτησης πρώτα σε πλάτος εξετάζει πρώτα όλες τις καταστάσεις που βρίσκονται στο ίδιο βάθος και μετά συνεχίζει στην επέκταση καταστάσεων στο αμέσως επόμενο επίπεδο.

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.
 - i. Αν η κατάσταση ανήκει στο κλειστό σύνολο τότε πήγαινε στο βήμα 2.
 - ii. Αν η κατάσταση είναι μία τελική τότε ανέφερε τη λύση. Αν θέλεις & άλλες λύσεις πήγαινε στο βήμα 2. Αλλιώς σταμάτησε.
4. Εφάρμοσε τους τελεστές μετάβασης για να βρεις τις καταστάσεις-παιδιά.
5. Βάλε τις καταστάσεις-παιδιά στο τέλος του μετώπου της αναζήτησης.
6. Βάλε την κατάσταση-γονέα στο κλειστό σύνολο.
7. Πήγαινε στο βήμα 2.

Αναζήτηση πρώτα κατά πλάτος



- Πλήρης, βέλτιστος
- Χωρική, χρονική πολυπλοκότητα $O(b^{d+1})$
 - π.χ. για $b=10$, 10.000 κόμβοι/sec, 1000 bytes/κόμβος:

Βάθος	Κόμβοι	Χρόνος	Μνήμη
2	1100	0.11 δευτερόλεπτα	1 megabyte
4	111.100	11 δευτερόλεπτα	106 megabytes
6	10^7	19 λεπτά	10 gigabytes
8	10^9	31 ώρες	1 terabyte
10	10^{11}	129 ημέρες	101 terabytes
12	10^{13}	35 χρόνια	10 petabytes
14	10^{15}	3.523 χρόνια	1 exabyte

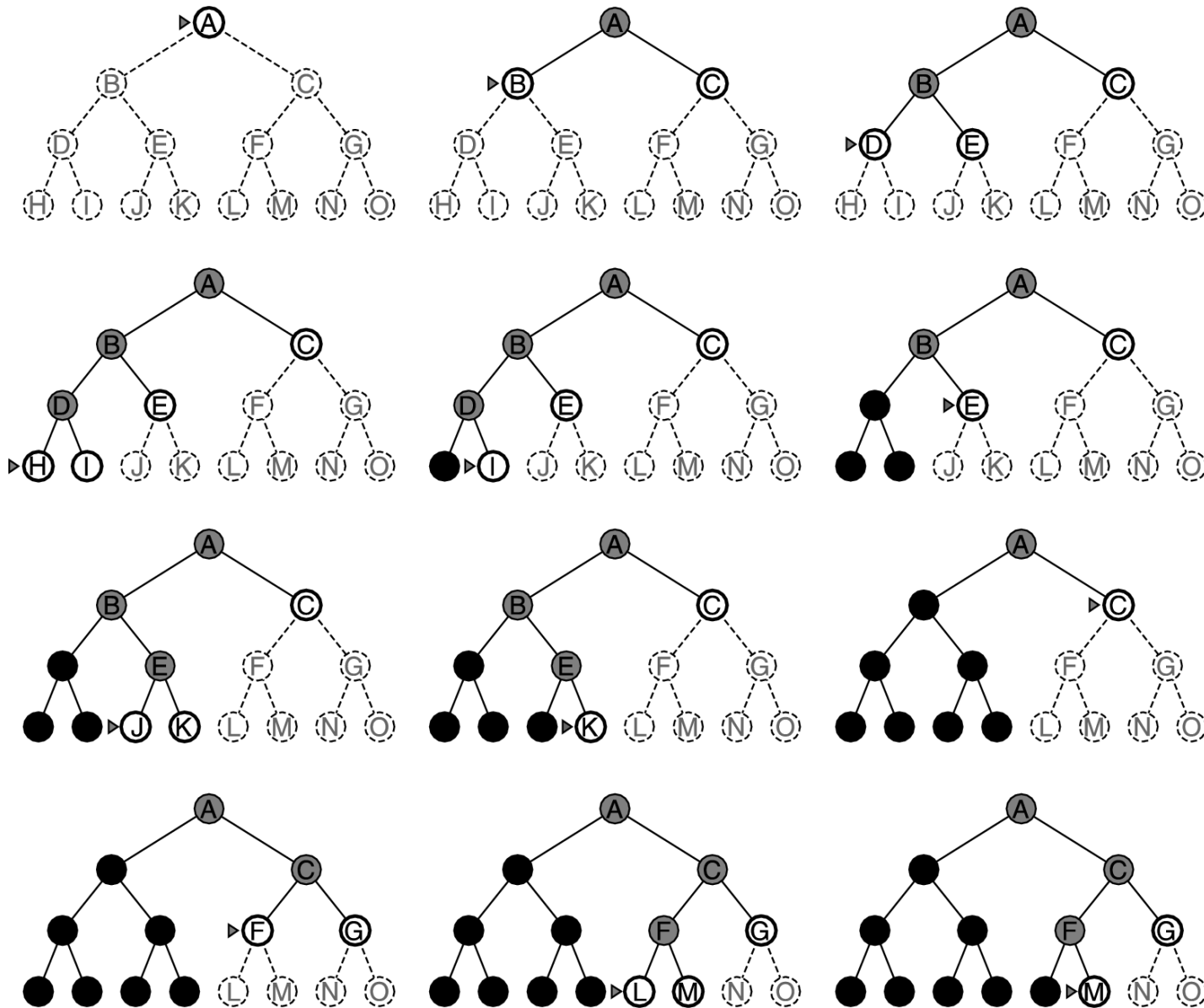
Αναζήτηση πρώτα κατά βάθος (Depth-first search)

Ο αλγόριθμος πρώτα σε βάθος επιλέγει προς επέκταση την κατάσταση που βρίσκεται πιο βαθιά στο δένδρο.

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.
 - i. Αν η κατάσταση ανήκει στο κλειστό σύνολο τότε πήγαινε στο βήμα 2.
 - ii. Αν η κατάσταση είναι μία από τις τελικές, τότε ανέφερε τη λύση. Αν θέλεις και άλλες λύσεις πήγαινε στο βήμα 2. Αλλιώς σταμάτησε.
4. Εφάρμοσε τους τελεστές μετάβασης για να βρεις τις καταστάσεις-παιδιά.
5. Βάλε τις καταστάσεις-παιδιά στην αρχή του μετώπου της αναζήτησης.
6. Βάλε την κατάσταση-γονέα στο κλειστό σύνολο.
7. Πήγαινε στο βήμα 2.

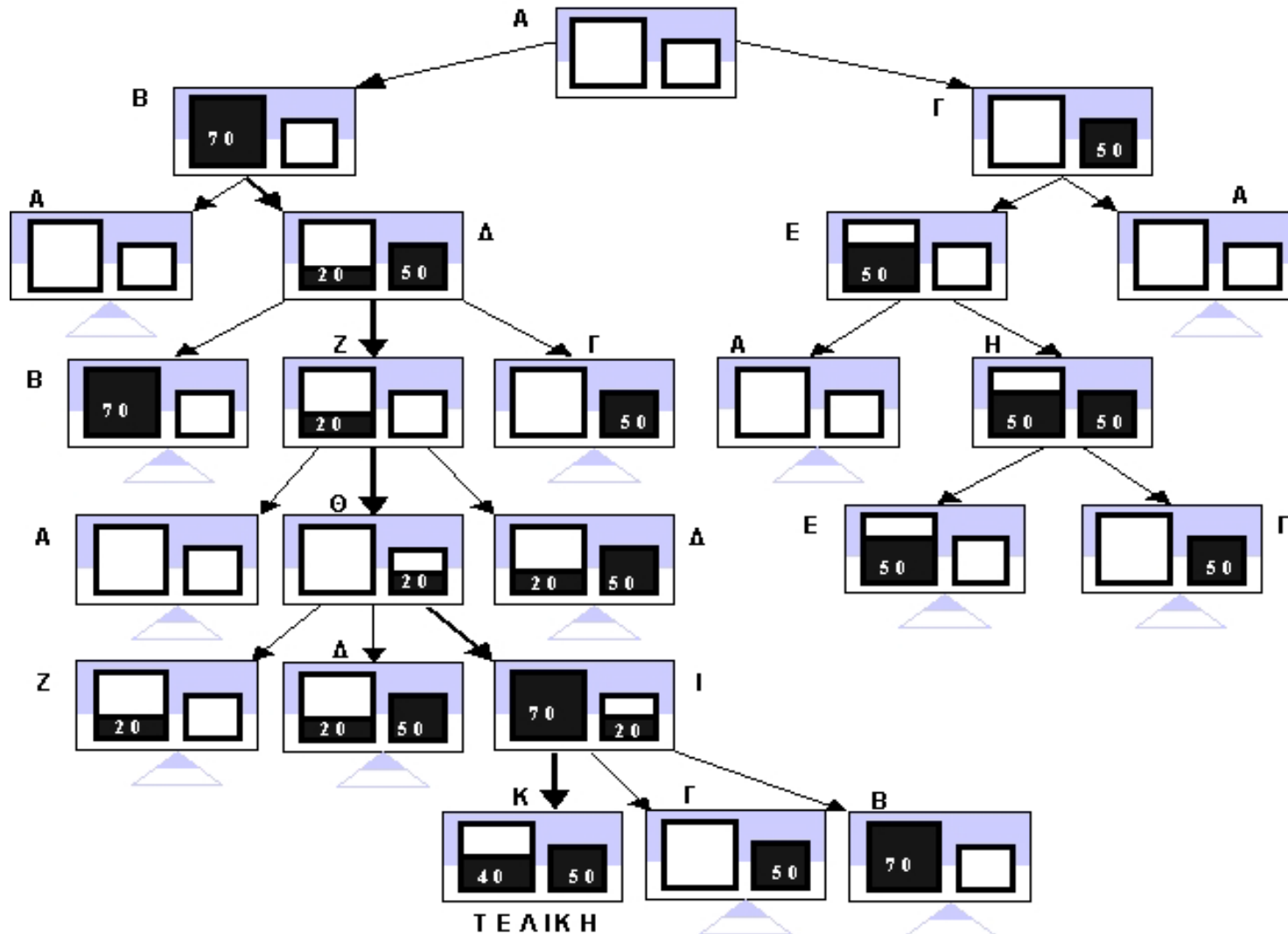
Αναζήτηση πρώτα κατά βάθος

(Depth-first search)



Αναζήτηση Πρώτα κατά Βάθος (DFS)

Δένδρο αναζήτησης στο πρόβλημα των ποτηριών



Αναζήτηση Πρώτα κατά Βάθος - Σχόλια

- Ελάχιστες απαιτήσεις σε χώρο: $b \cdot m + 1$
- Πολυπλοκότητα χρόνου: $O(b^m)$
- Πλεονεκτήματα:
 - Το μέτωπο της αναζήτησης δε μεγαλώνει πάρα πολύ.
 - Μικρές απαιτήσεις σε χώρο: $b \cdot m + 1$
- Μειονεκτήματα:
 - Δεν εγγυάται ότι η πρώτη λύση που θα βρεθεί είναι η βέλτιστη.
 - Θεωρείται μη-πλήρης.
 - Όταν όμως ο χώρος αναζήτησης είναι πεπερασμένος ο DFS θα βρει λύση, εάν μια τέτοια υπάρχει.

Αναζήτηση περιορισμένου βάθους

(Depth-limited search)

- Τίθεται όριο βάθους $L < m$:
 - Οι κόμβοι σε βάθος L αντιμετωπίζονται σαν να μην έχουν διαδόχους
- Λύνεται το πρόβλημα των άπειρων διαδρομών
- Απώλεια πληρότητας αν $L < d$:
 - Αν ο ρηχότερος κόμβος στόχου βρίσκεται πέρα από το όριο βάθους L
- Όχι βέλτιστες λύσεις αν $L > d$
- Χρονική & Χωρική πολυπλοκότητα: $O(b^L)$
- Επιλογή L με γνώση του προβλήματος

Αλγόριθμος Επαναληπτικής Εκβάθυνσης

- Ο αλγόριθμος επαναληπτικής εκβάθυνσης (Iterative Deepening - ID) συνδυάζει με τον καλύτερο τρόπο τους DFS και BFS.
 1. Όρισε το αρχικό βάθος αναζήτησης (συνήθως 1).
 2. Εφάρμοσε τον αλγόριθμο DFS μέχρι αυτό το βάθος αναζήτησης.
 3. Αν έχεις βρει λύση σταμάτησε.
 4. Αύξησε το βάθος αναζήτησης (συνήθως κατά 1).
 5. Πήγαινε στο βήμα 2.

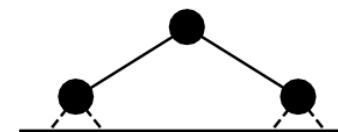
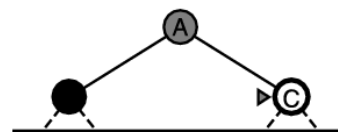
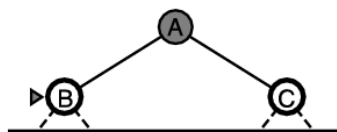
Επαναληπτική εκβάθυνση (1/3)

(Iterative-deepening search)

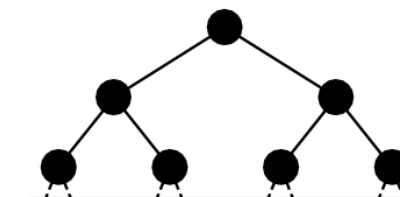
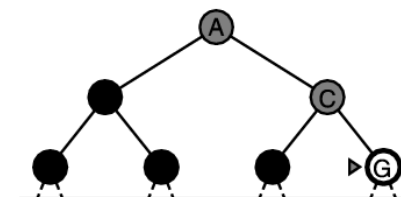
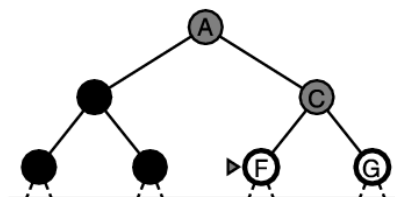
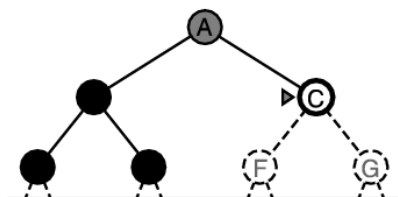
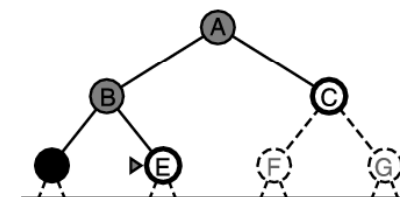
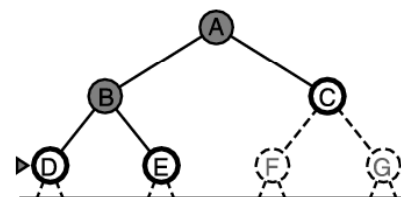
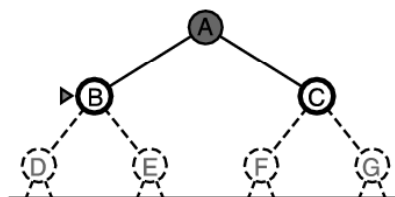
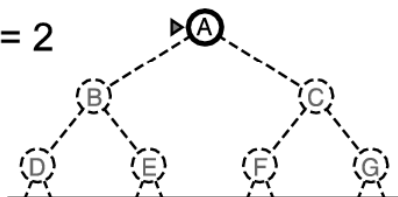
Όριο = 0



Όριο = 1



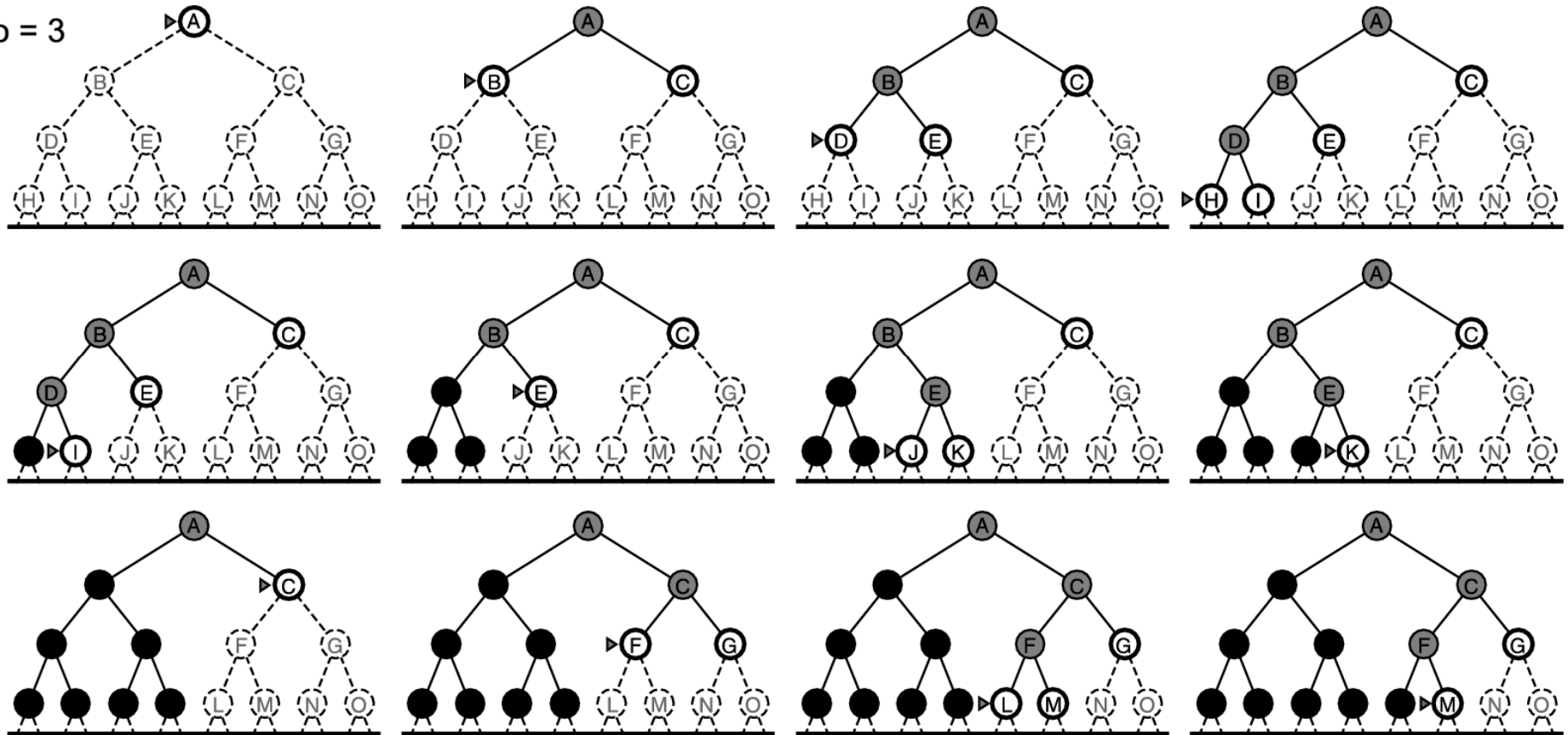
Όριο = 2



Επαναληπτική εκβάθυνση (2/3)

(Iterative-deepening search)

Όριο = 3



Επαναληπτική εκβάθυνση (3/3)

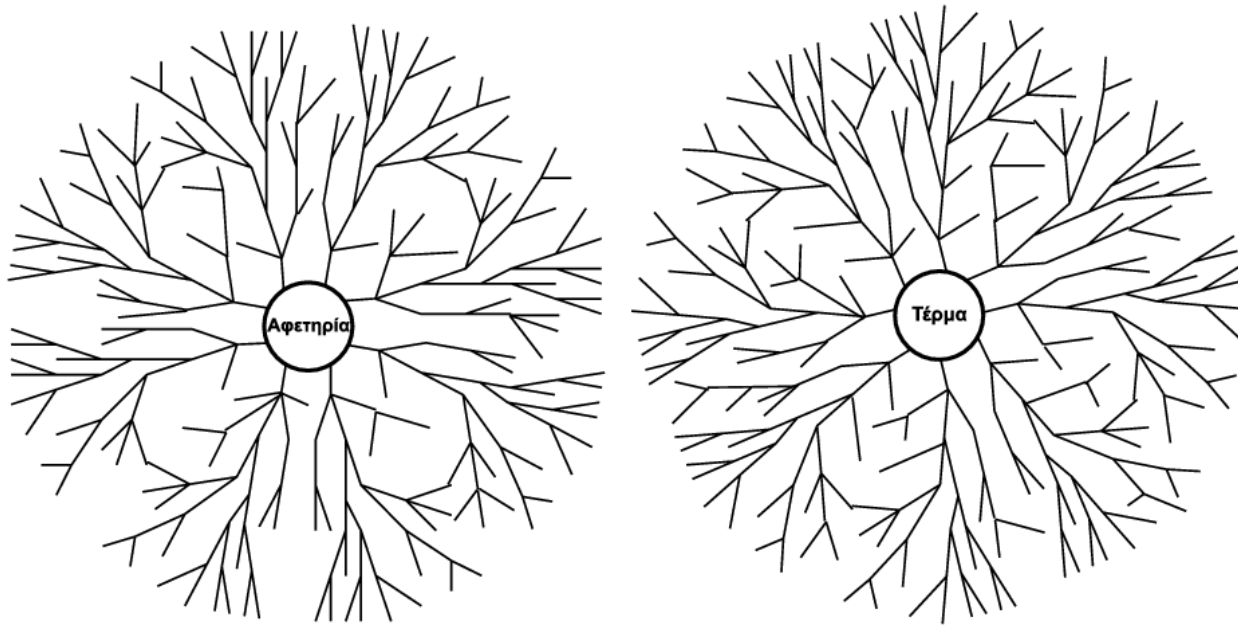
(Iterative-deepening search)

- Μειονέκτημά το ότι δεν θυμάται τίποτα από τον προηγούμενο κύκλο αναζήτησης.
- Χωρική πολυπλοκότητα: $O(bd)$
- Χρονική πολυπλοκότητα: $O(b^d)$
- Είναι πλήρης όταν ο παράγοντας διακλάδωσης είναι πεπερασμένος
- Είναι βέλτιστη όταν το κόστος διαδρομής είναι μη φθίνουσα συνάρτηση του βάθους του κόμβου
- Γενικά, η επαναληπτική εκβάθυνση είναι η προτιμότερη μέθοδος απληροφόρητης αναζήτησης όταν ο χώρος αναζήτησης είναι μεγάλος και το βάθος της λύσης δεν είναι γνωστό.

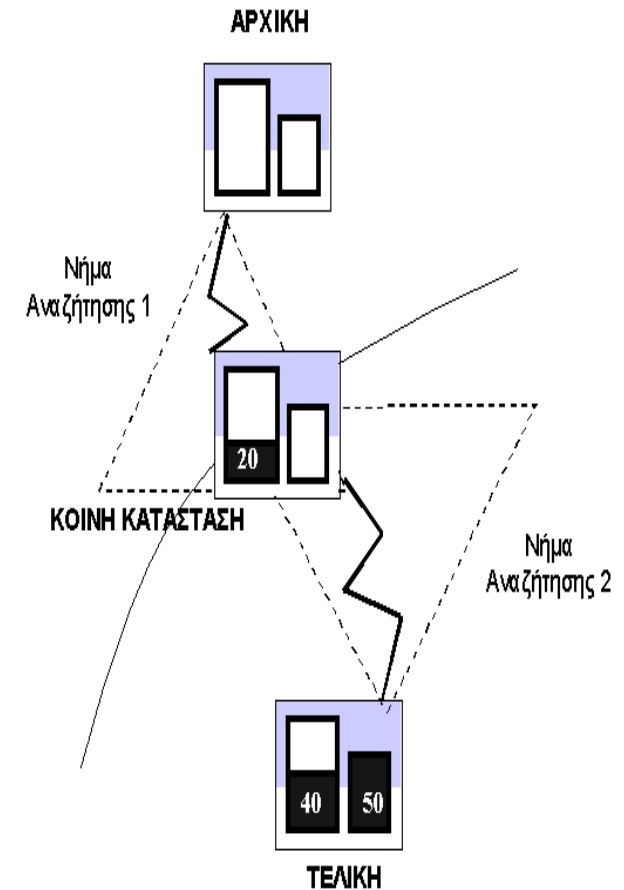
Αναζήτηση Διπλής Κατεύθυνσης (Αμφίδρομη)

- Η ιδέα της αναζήτησης διπλής κατεύθυνσης πηγάζει από τη δυνατότητα του παραλληλισμού στα υπολογιστικά συστήματα.
- Προϋποθέσεις κάτω από τις οποίες μπορεί να εφαρμοστεί:
 - Οι τελεστές μετάβασης είναι αντιστρέψιμοι (reversible), και
 - Είναι πλήρως γνωστή η τελική κατάσταση.
- Λειτουργία:
 - Αρχίζει την αναζήτηση από την αρχική και τελική κατάσταση ταυτόχρονα.
 - Αν κάποια κατάσταση που επεκτείνεται είναι κοινή, τότε βρέθηκε λύση.
 - Λύση είναι η ένωση των μονοπατιών από την κοινή κατάσταση έως την αρχική και έως την τελική κατάσταση.
- Μειονεκτήματα:
 - Υπάρχει επιπλέον κόστος επικοινωνίας μεταξύ των δύο αναζητήσεων.

Αναζήτηση Διπλής Κατεύθυνσης

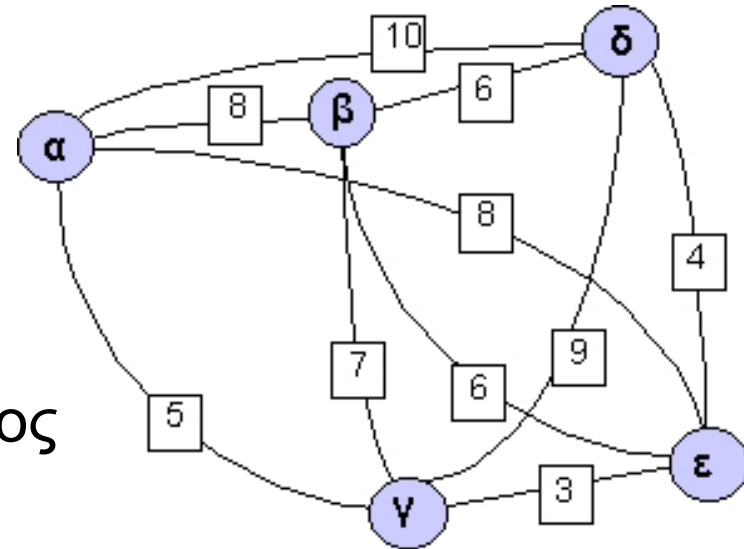


- Χρονική πολυπλοκότητα: $O(b^{d/2})$
- Χωρική πολυπλοκότητα: $O(b^{d/2})$
- Προβλήματα:
 - Έλλειψη καλά καθορισμένων στόχων (π.χ. σκάκι)



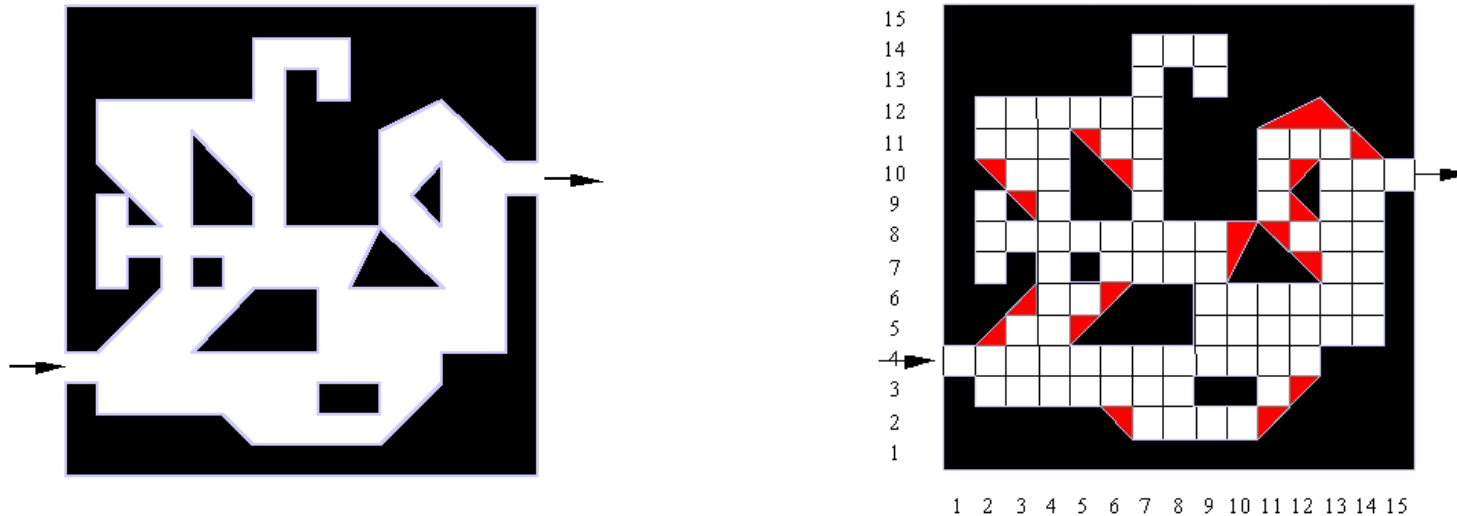
Επέκταση και Οριοθέτηση (Branch and Bound B&B)

- Ο B&B κλαδεύει καταστάσεις (pruning) και μειώνει το χώρο αναζήτησης.
- π.χ.: βρέθηκε μια λύση με κόστος 159. Κατά την αναζήτηση για άλλες λύσεις μια κατάσταση έχει ήδη κόστος 167. Άρα κλαδεύεται αυτή η κατάσταση.
- Ο B&B εφαρμόζεται όταν υπάρχει μια πραγματική εκτίμηση του κόστους.
- Το κέρδος από το κλάδεμα εξαρτάται από το πόσο γρήγορα θα βρεθεί μια καλή λύση.
- Υπάρχει περίπτωση να μη γίνει καθόλου κλάδεμα αν οι λύσεις είναι διατεταγμένες από τη χειρότερη προς την καλύτερη.
- Στη χειρότερη περίπτωση συμπεριφέρεται σαν τον DFS.



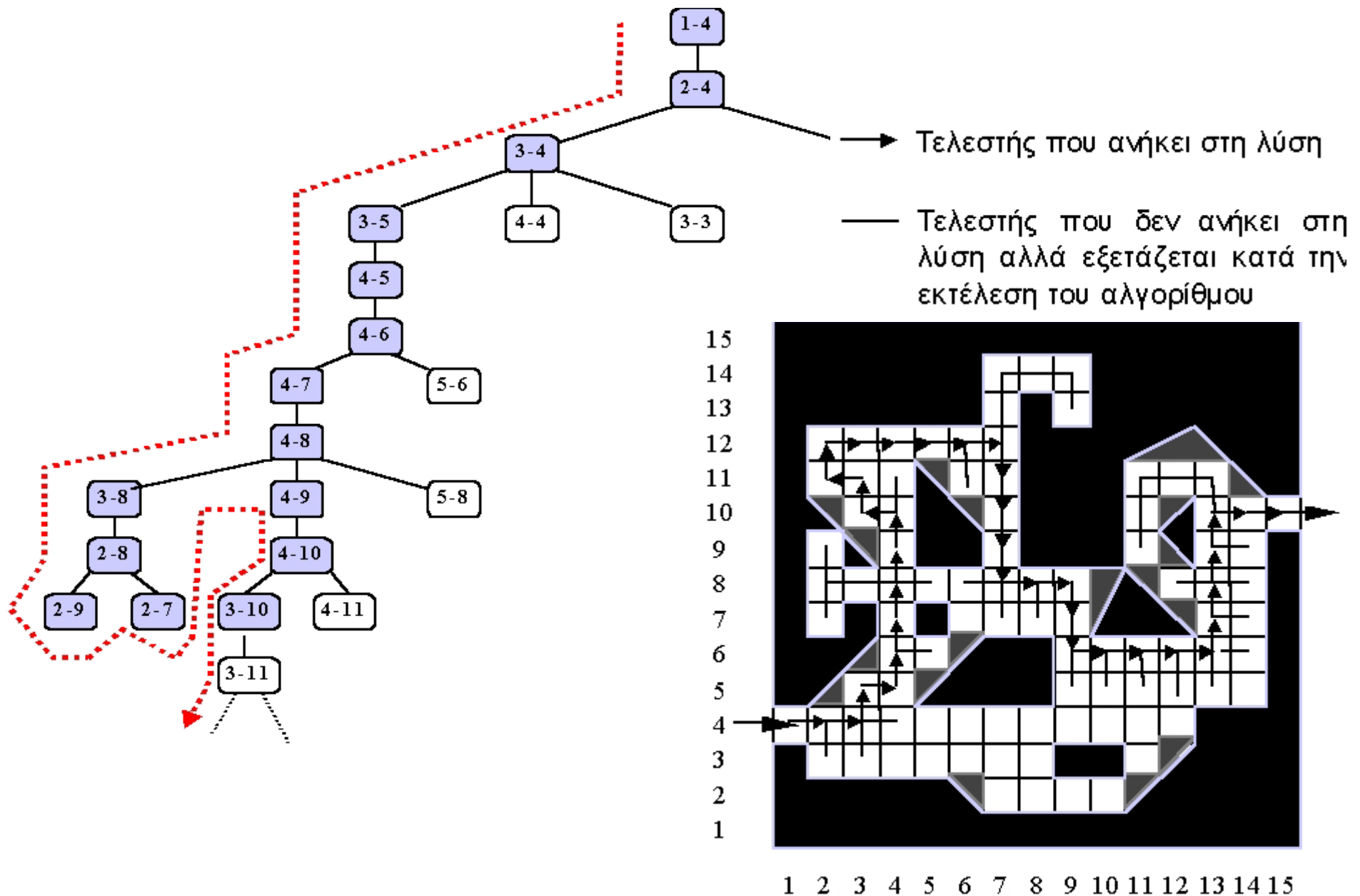
Εφαρμογή των Αλγορίθμων Τυφλής Αναζήτησης

Το πρόβλημα του Λαβύρινθου - Ορισμός

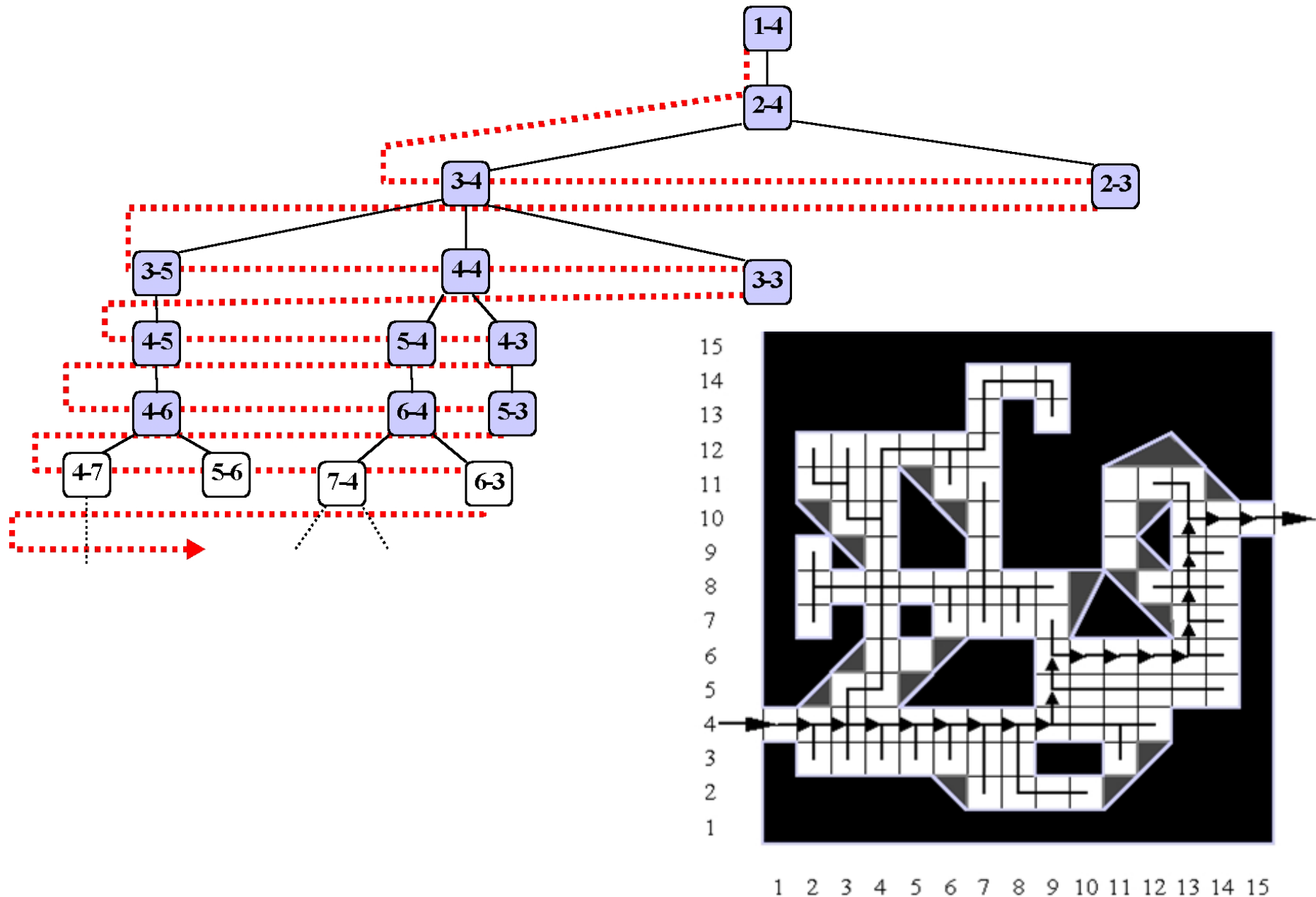


- Αρχική κατάσταση (1,4).
- Το σύνολο τελικών καταστάσεων περιέχει μόνο (15,10).
- Οι τελεστές μεταφοράς είναι οι εξής (εφόσον η θέση είναι ελεύθερη):
 - πηγαινε μία θέση αριστερά,
 - πηγαινε μία θέση επάνω,
 - πηγαινε μία θέση δεξιά,
 - πηγαινε μία θέση κάτω.
- Ο χώρος καταστάσεων είναι όλες οι ελεύθερες θέσεις, χωρίς εμπόδια, του πλέγματος.

Εφαρμογή DFS στο πρόβλημα του λαβύρινθου

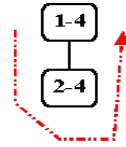


Εφαρμογή BFS στο πρόβλημα του λαβύρινθου

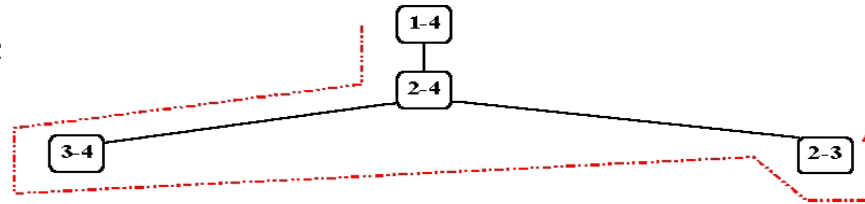


Εφαρμογή του ID στο πρόβλημα του λαβυρίνθου

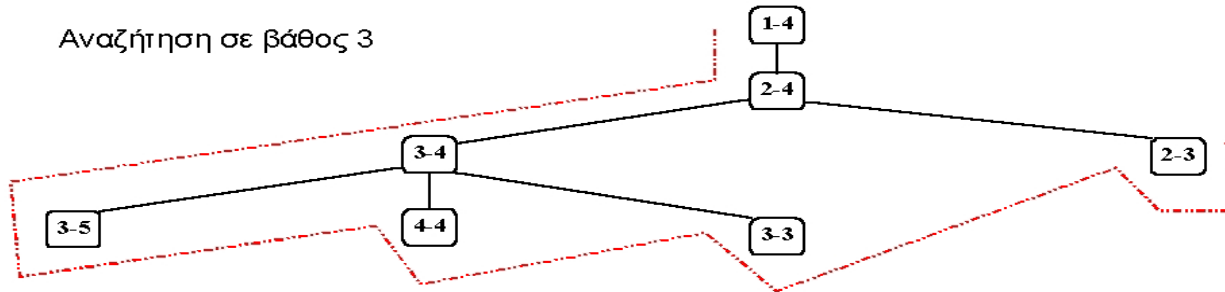
Αναζήτηση σε βάθος 1



Αναζήτηση σε βάθος 2



Αναζήτηση σε βάθος 3



Αναζήτηση σε βάθος 4

