



Τεχνητή Νοημοσύνη

3η διάλεξη (2025-26)

Ίων Ανδρουτσόπουλος

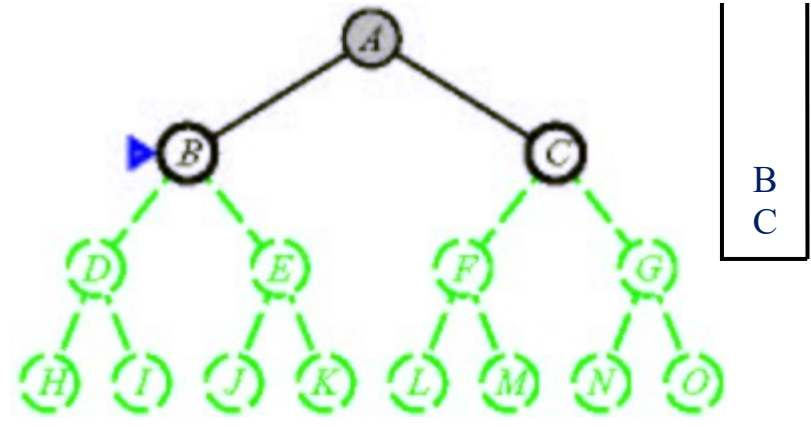
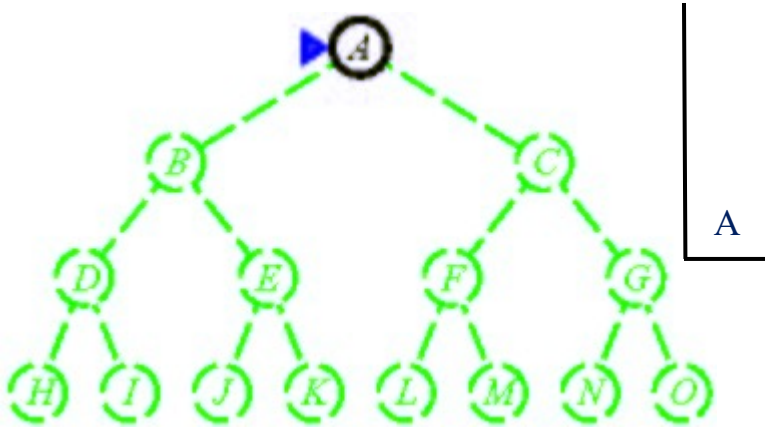
<http://www.aueb.gr/users/ion/>

Οι διαφάνειες αυτής της διάλεξης βασίζονται στα βιβλία *Τεχνητή Νοημοσύνη των Βλαχάβα κ.ά.*, 3η έκδοση, Β. Γκιούρδας Εκδοτική, 2006 και *Artificial Intelligence – A Modern Approach* των S. Russel και P. Norvig, 2^η και 4^η έκδοση, Prentice Hall, 2003 και 2020. Τα σχήματα των διαφανειών προέρχονται από αντίστοιχες διαφάνειες των δύο βιβλίων.

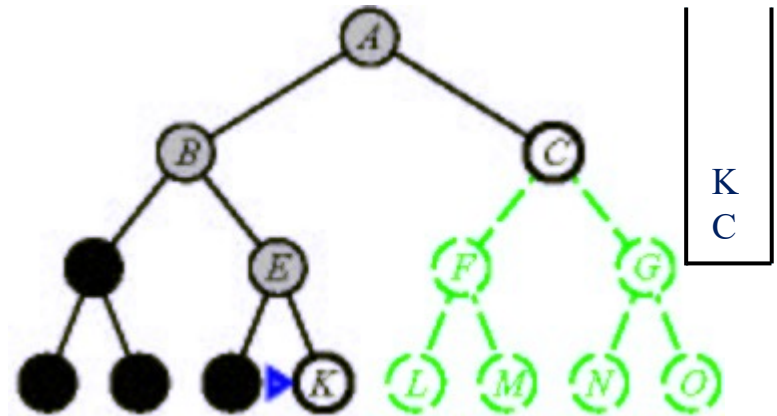
Τι θα ακούσετε σήμερα

- Αναζήτηση πρώτα σε βάθος και παραλλαγές.
- Επαναληπτική εκβάθυνση.
- Ευρετικές συναρτήσεις.
- Αναζήτηση πρώτα του καλύτερου.

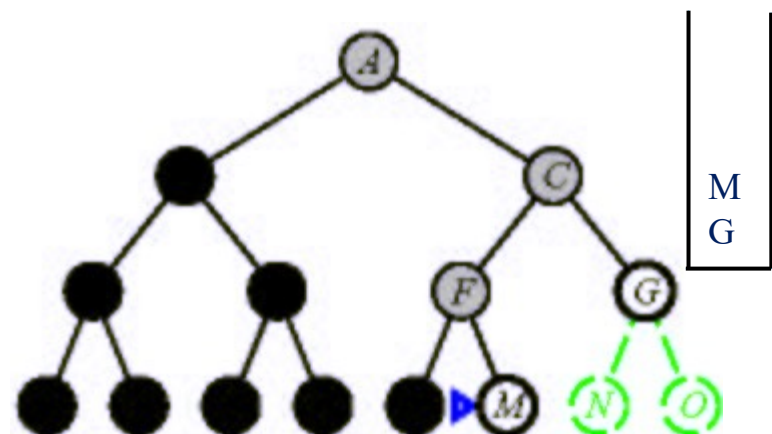
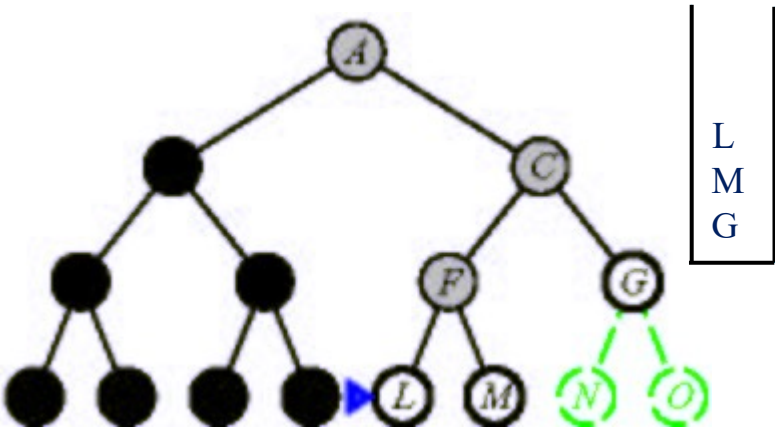
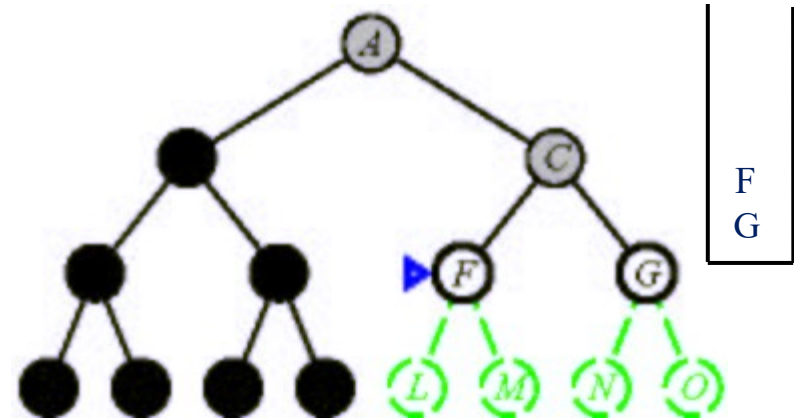
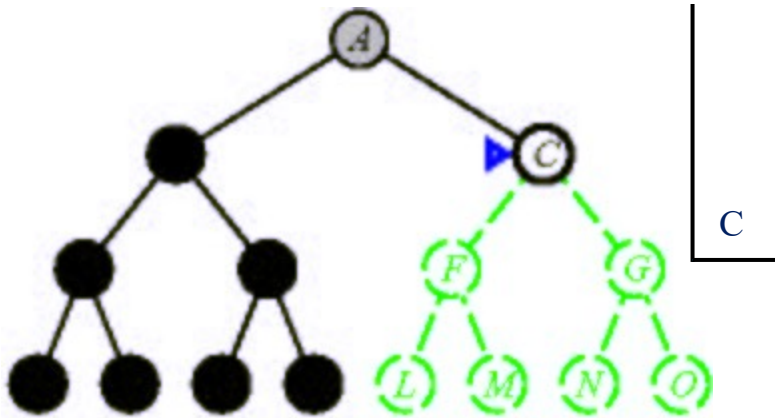
Αναζήτηση πρώτα σε βάθος (DFS)



Αναζήτηση πρώτα σε βάθος (DFS)



Αναζήτηση πρώτα σε βάθος (DFS)



Αναζήτηση πρώτα σε βάθος (DFS)

1. Βάλτε τη **ρίζα** (κόμβος αρχικής κατάστασης) στο μέτωπο αναζήτησης.
2. Αν το **μέτωπο** είναι **άδειο**, σταμάτα.
3. Βγάλε τον **πρώτο** σε σειρά κόμβο από το μέτωπο.
4. Αν ο κόμβος αντιστοιχεί σε **τελική κατάσταση**, επέστρεψε τη λύση.
5. **Επέκτεινε** τον κόμβο και πρόσθεσε τα παιδιά στην **αρχή** του μετώπου αναζήτησης (**στοίβα**).
6. Πήγαινε στο βήμα 2.

Χαρακτηριστικά του DFS

- **Μη πλήρης:** μπορεί να παγιδευθεί σε **άπειρα κλαδιά**.
- **Μη βέλτιστος.**
 - Θεωρούμε ότι το κόστος λύσεως είναι αύξουσα συνάρτηση του βάθους (και μόνο).
 - Μπορεί να **μη βρει** μια **εναλλακτική** τελική κατάσταση σε **μικρότερο βάθος**.
- **Χρονική πολυπλοκότητα: $O(b^m)$.**
 - Στη χειρότερη περίπτωση παράγουμε όλους τους κόμβους του δέντρου αναζήτησης: $b + b^2 + b^3 + \dots + b^m = O(b^m)$.
 - Αν θέλουμε να **λάβουμε υπόψη μας και το βάθος**, έστω d' , της λύσης που βρίσκουμε, στη χειρότερη περίπτωση η τελική κατάσταση της λύσης που βρίσκουμε είναι ο δεξιότερος κόμβος βάθους d' . Παράγουμε όλους τους κόμβους του δέντρου αναζήτησης εκτός από τους κόμβους του υποδέντρου της τελικής κατάστασης. $O(b^m) - O(b^{m-d'}) = O(b^m)$.

Χαρακτηριστικά του DFS – συνέχεια

- Πολυπλοκότητα χώρου: $O(bm)$.
 - Αποθηκεύουμε το **μέτωπο** (άσπροι) και τους **προγόνους των κόμβων του μετώπου** (γκρίζοι, για να μπορούμε να επιστρέψουμε τη λύση).
 - Η στιγμή όπου χρειαζόμαστε την περισσότερη μνήμη είναι όταν εξετάζουμε το αριστερότερο φύλλο σε βάθος m : αποθηκεύουμε τα **παιδιά όλων των προγόνων του φύλλου και τη ρίζα**, δηλαδή $mb + 1$ κόμβους.
 - Κανένα ουσιώδες όφελος αν αποθηκεύουμε **μόνο το μέτωπο**: στη χειρότερη στιγμή αποθηκεύουμε τα $b - 1$ **παιδιά όλων των προγόνων του φύλλου και το ίδιο το φύλλο**, δηλαδή $m \cdot (b - 1) + 1$ κόμβους. Πάλι $O(bm)$.

DFS με οπισθοδρόμηση

- Παράγουμε **μόνο ένα παιδί** όποτε επεκτείνουμε έναν κόμβο.
- **Οπισθοδρόμηση:**
 - Αν φτάσουμε σε αδιέξοδο, επιστρέφουμε στον πατέρα (ή αναδρομικά στους προγόνους) και ζητούμε να παραχθεί το επόμενο παιδί.
- Η πολυπλοκότητα **χώρου** μειώνεται σε **$O(m)$** .
 - Δεν χρειάζεται να αποθηκεύουμε το μέτωπο, αλλά **μόνο τους προγόνους**.
- Ίδια χρονική πολυπλοκότητα: **$O(b^m)$** .

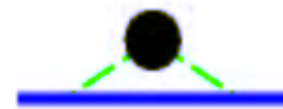
DFS με περιορισμό βάθους

- Όπως ο DFS, αλλά με **μέγιστο επιτρεπτό βάθος (l)**.
 - Τους κόμβους που βρίσκονται στο μέγιστο επιτρεπτό βάθος τους αντιμετωπίζουμε σαν να μην έχουν παιδιά.
 - **Αποφεύγουμε τα άπειρα κλαδιά.**
 - Όμως δεν είναι πάντα εύκολο να διαλέξουμε το l .
- **Μη πλήρης.**
 - Τώρα μπορεί να αποτύχει να βρει λύση λόγω του περιορισμού βάθους.
- **Μη βέλτιστος**, όπως ο απλός DFS.
- **Χρονική πολυπλοκότητα: $O(b^l)$.**
- **Πολυπλοκότητα χώρου: $O(bl)$.**

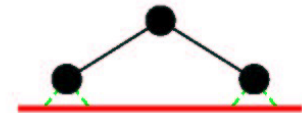
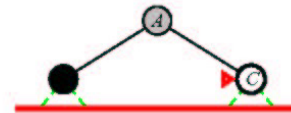
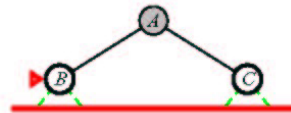
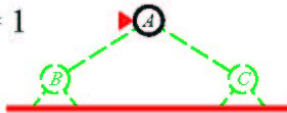
Επαναληπτική εκβάθυνση (IDS)

1. Θέσε το μέγιστο βάθος αναζήτησης σε 0.
2. Εφάρμοσε τον **DFS** μέχρι το μέγιστο βάθος αναζήτησης.
3. Αν βρέθηκε λύση, επίστρεψε τη λύση.
4. **Αύξησε το μέγιστο βάθος αναζήτησης κατά 1.**
5. Πήγαινε στο βήμα 2.

Limit = 0

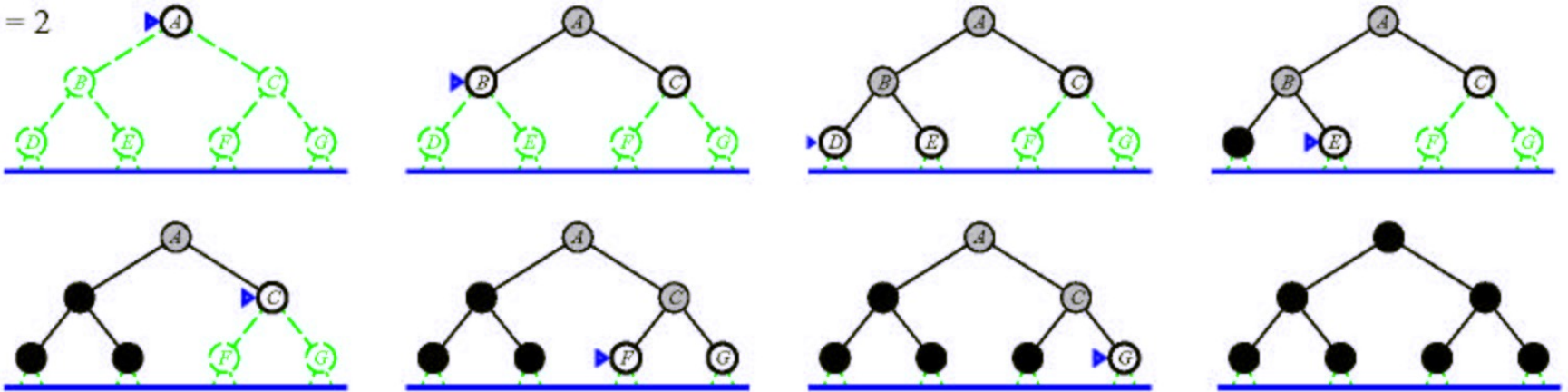


Limit = 1

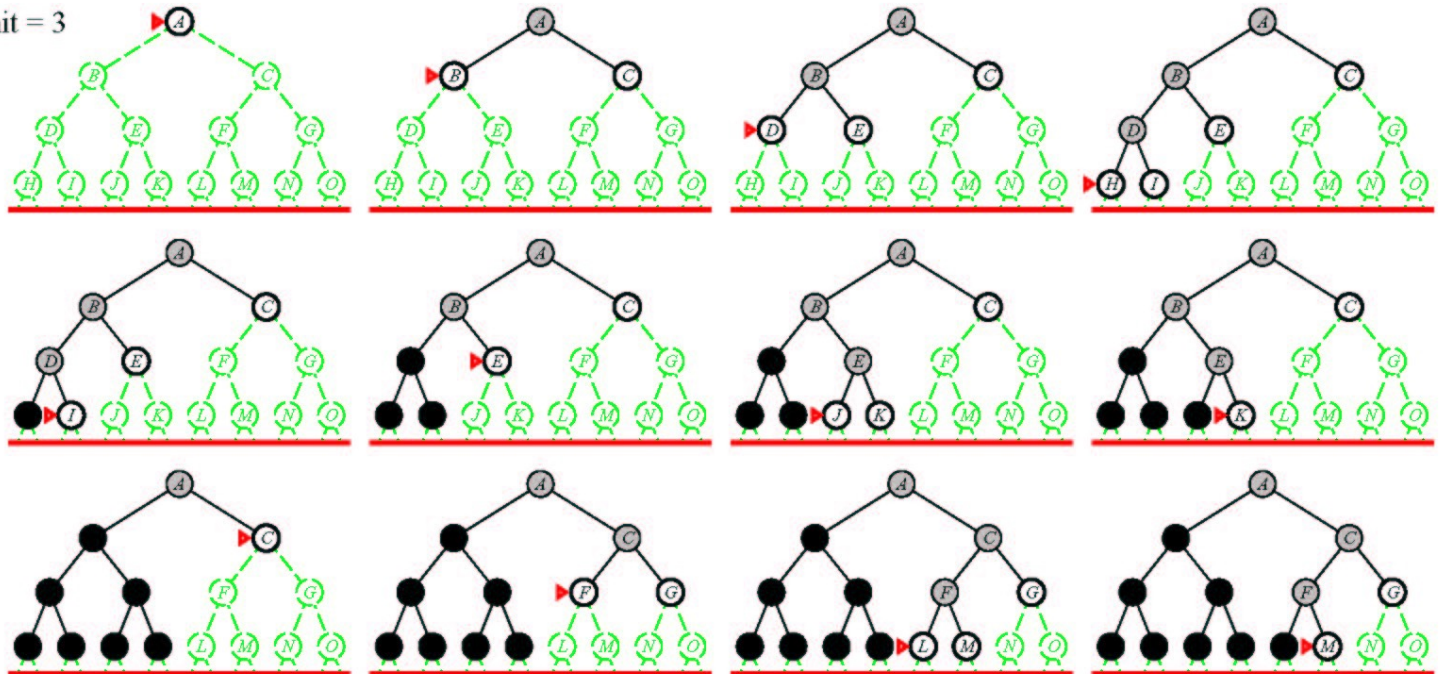


Επαναληπτική εκβάθυνση (IDS)

Limit = 2



Limit = 3



Χαρακτηριστικά του IDS

- **Πλήρης**, αν το b είναι πεπερασμένο, όπως ο BFS.
- **Βέλτιστος**, αν το κόστος λύσεως είναι αύξουσα συνάρτηση του βάθους (και μόνο), όπως ο BFS.
- Πολυπλοκότητα χώρου: $O(bd)$.
 - Όπως στον DFS, αλλά τώρα το **μέγιστο βάθος** στο οποίο φτάνουμε είναι d .
 - Δεν θα υπερβούμε ποτέ το βάθος d της ρηχότερης λύσης.
- **Χρονική πολυπλοκότητα: $O(b^d)$** .
 - Συνολικά $d + 1$ επαναλήψεις (ξεκινάμε από μέγιστο βάθος 0).
 - **Οι κόμβοι βάθους h παράγονται $d - (h - 1)$ φορές.** (Η ρίζα δεν παράγεται.)
 - $d \cdot b + (d-1) \cdot b^2 + (d-2) \cdot b^3 + \dots + (2) \cdot b^{d-1} + (1) \cdot b^d = O(b^d)$.
- Γενικά **ο καλύτερος** αλγόριθμος τυφλής αναζήτησης.
 - Αλλά ακόμα και αυτός έχει **εκθετική** χρονική πολυπλοκότητα.

Κλειστό σύνολο

- Αποθηκεύουμε στη μνήμη το **κλειστό σύνολο**:
 - Περιέχει όλες τις καταστάσεις (όχι τους κόμβους του δένδρου αναζήτησης) που έχουμε «συναντήσει».
- **Δεν επεκτείνουμε** καταστάσεις που βρίσκονται ήδη στο κλειστό σύνολο.
 - Στον DFS αποφεύγουμε έτσι τα άπειρα μονοπάτια που αντιστοιχούν σε **κύκλους** του γράφου αναζήτησης.
 - Δεν αποφεύγουμε όμως τα υπόλοιπα άπειρα μονοπάτια που ενδέχεται να υπάρχουν σε προβλήματα με **άπειρες καταστάσεις** (π.χ. $x = 1 \rightarrow x = 2 \rightarrow x = 3 \rightarrow \dots$).
- Με κλειστό σύνολο, **χάνεται η γραμμική πολυπλοκότητα χώρου** των DFS και IDS.
 - Στη χειρότερη περίπτωση, όπου όλοι οι κόμβοι αντιστοιχούν σε διαφορετικές καταστάσεις, αποθηκεύουμε στο κλειστό σύνολο όλους τους κόμβους που παράγουμε.
 - $O(b^m)$ στον DFS και $O(b^d)$ στον IDS.

Τυφλή και ευρετική αναζήτηση

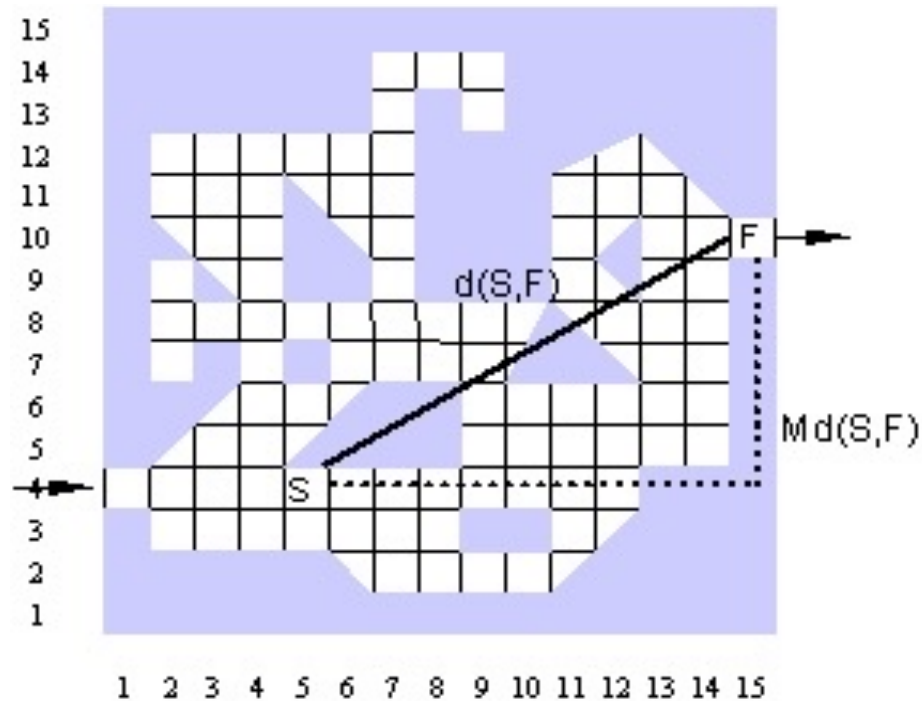
- **Αλγόριθμοι τυφλής αναζήτησης:**
 - Καμία πρόβλεψη για το αν το μονοπάτι που εξερευνούμε οδηγεί σε τελική κατάσταση, ούτε πρόβλεψη κόστους μονοπατιού.
 - Π.χ. DFS, BFS, IDS.
- **Αλγόριθμοι ευρετικής αναζήτησης:**
 - Η αναζήτηση καθοδηγείται από έναν «ευρετικό» μηχανισμό, που αξιολογεί την κάθε δυνατή κατεύθυνση.
 - Ο ευρετικός μηχανισμός ενσωματώνει γνώση για το συγκεκριμένο πρόβλημα.
 - Προτιμούμε κατευθύνσεις με υψηλή προβλεπόμενη αξία ή χαμηλότερο προβλεπόμενο κόστος λύσης.

Ευρετικός μηχανισμός

- Συνήθως μια **συνάρτηση** $h: S \rightarrow R$, που αξιολογεί κάθε κατάσταση.
 - **$h(s)$: εκτίμηση του κόστους του βέλτιστου μονοπατιού μεταβάσεων από την s ως μία τελική κατάσταση.**
 - Οι εκτιμήσεις είναι **προσεγγιστικές** και δεν οδηγούν πάντα στο σωστό συμπέρασμα.
 - Αλλά **αν η s είναι τελική κατάσταση**, πρέπει οπωσδήποτε **$h(s) = 0$** .
- Γενική ιδέα: **επεκτείνουμε** την κατάσταση s του μετώπου με το **μικρότερο $h(s)$** .

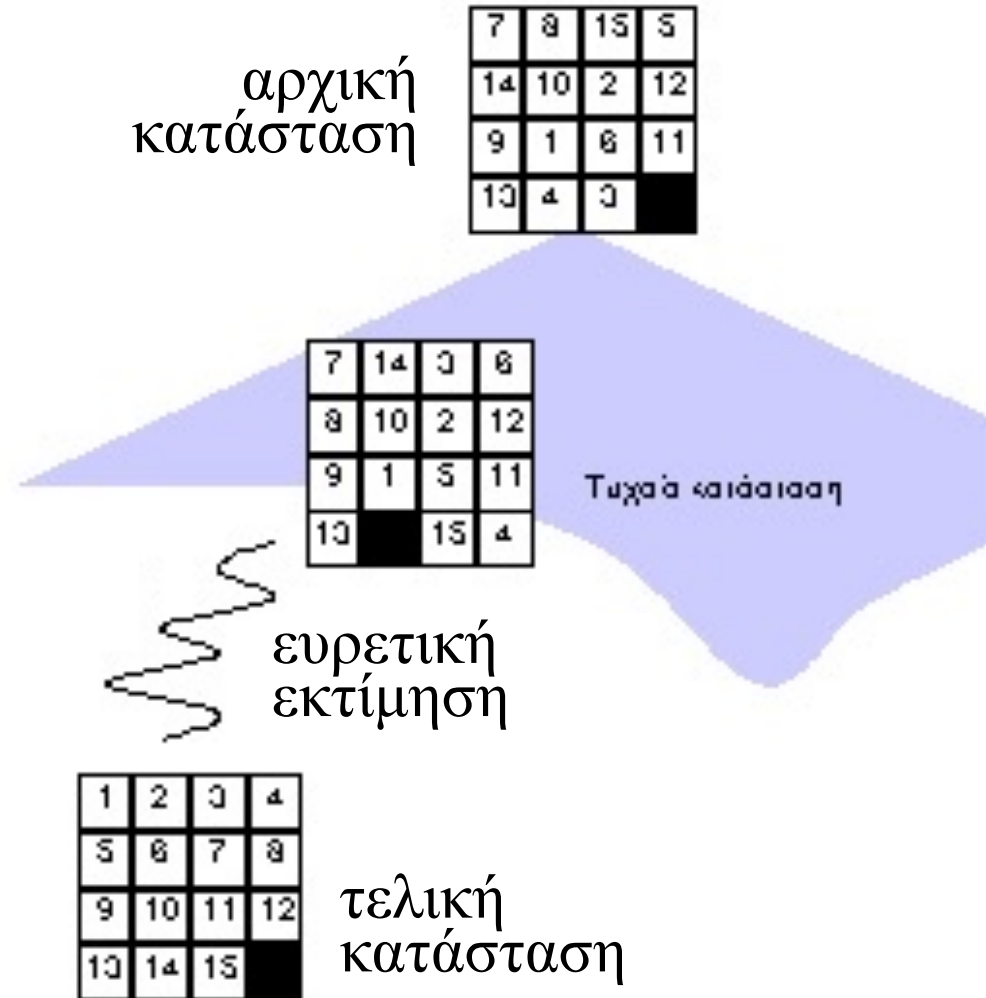
Ευρετικές συναρτήσεις

- Ευκλείδεια απόσταση: $d(S,F) = \sqrt{(X_S - X_F)^2 + (Y_S - Y_F)^2}$
- Απόσταση Manhattan: $Md(S,F) = |X_S - X_F| + |Y_S - Y_F|$



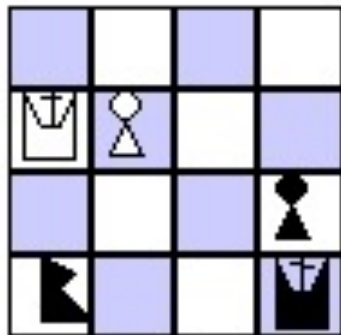
Ευρετικές συνάρτησεις

- $h_1(s)$: πόσα πλακίδια βρίσκονται **εκτός θέσης**;
- $h_2(s)$: **άθροισμα αποστάσεων** Manhattan των πλακιδίων από τις τελικές τους θέσεις.



Ευρετικές συναρτήσεις για σκάκι

- **Συνολική αξία κομματιών:** π.χ. βασιλιάς: 10, άλογο: 5, πiónι: 1.
- **Θέση:** Κάθε κομμάτι στα 4 κεντρικά τετράγωνα παίρνει επιπλέον 2 πόντους.
- **Απειλές:** Για κάθε απειλή: 3 επιπλέον πόντοι.
Απειλή προς βασιλιά: 20 πόντοι.



βασιλιάς



άλογο



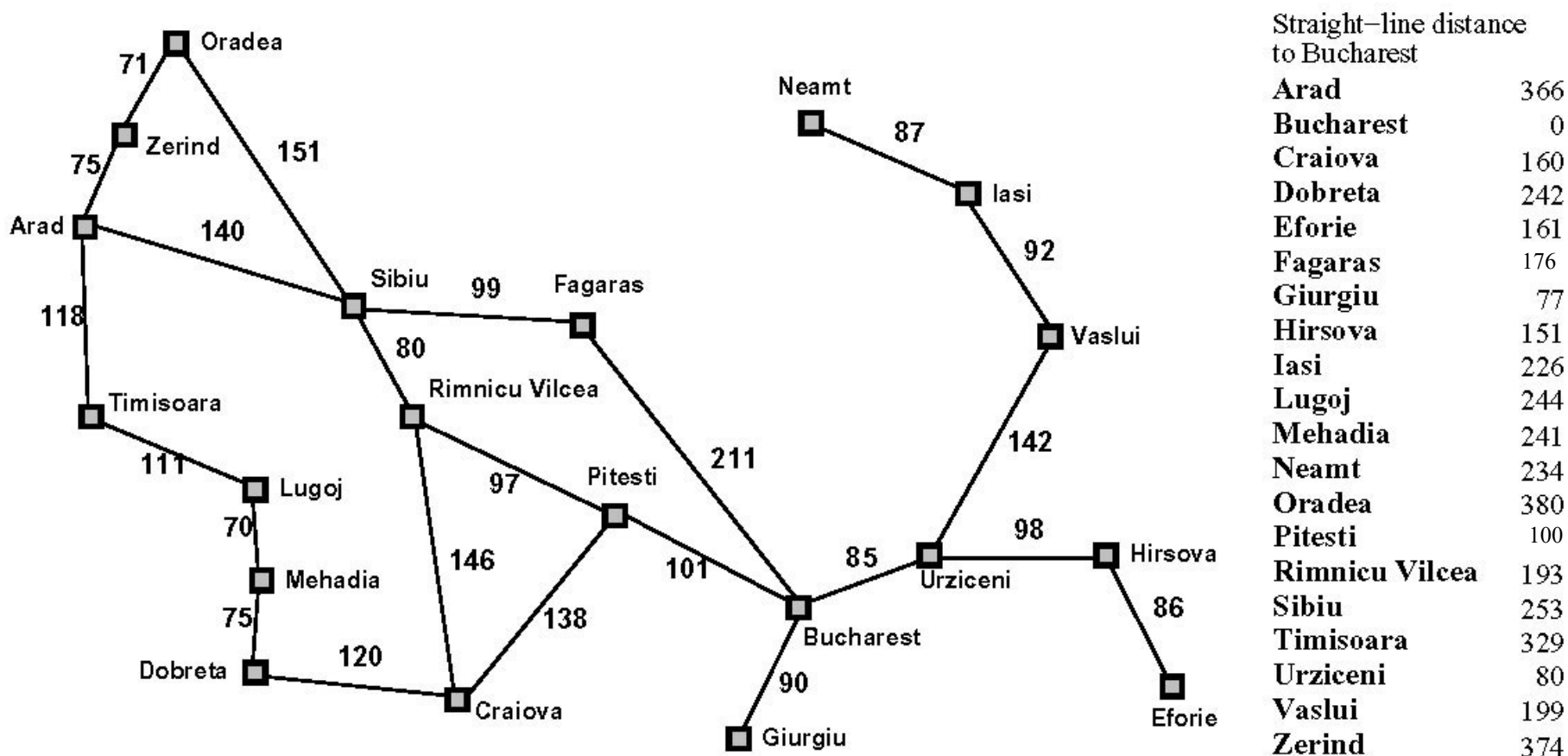
πiónι

Αναζήτηση πρώτα του καλύτερου (BestFS)

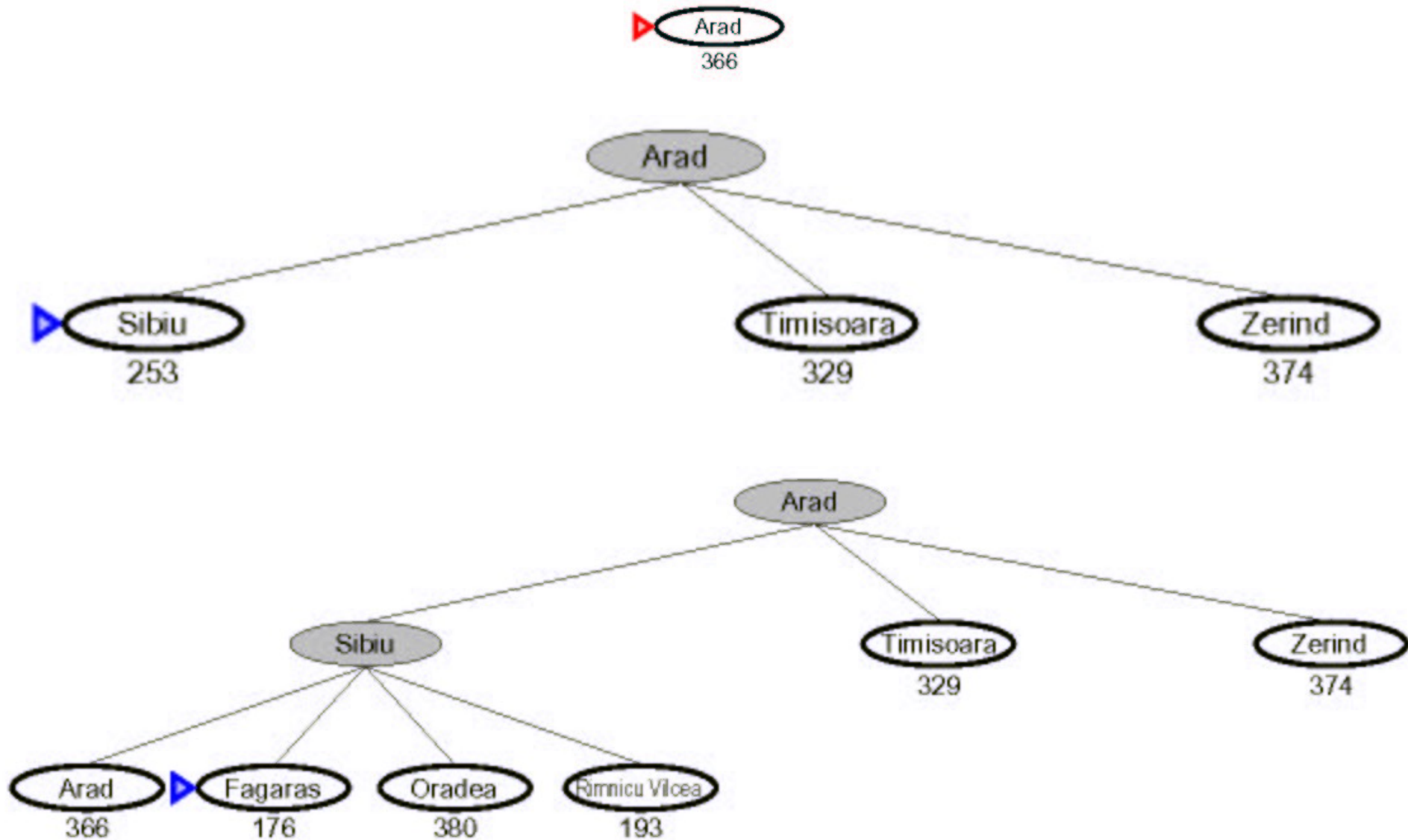
1. Βάλε τη **ρίζα** (κόμβος αρχικής κατάστασης) στο μέτωπο αναζήτησης.
2. Αν το **μέτωπο είναι άδειο**, σταμάτα.
3. Βγάλε τον **πρώτο σε σειρά** κόμβο από το μέτωπο.
4. Αν ο κόμβος αντιστοιχεί σε **τελική κατάσταση**, τύπωσε τη λύση και σταμάτα.
5. **Επέκτεινε** τον κόμβο και **πρόσθεσε τα παιδιά** στο μέτωπο αναζήτησης.
6. **Αναδιάταξε** το μέτωπο αναζήτησης **σύμφωνα με την ευρετική συνάρτηση**, ώστε οι κόμβοι των καλύτερων καταστάσεων να βρίσκονται στην αρχή.
7. Πήγαινε στο βήμα 2.

Αναζήτηση με BestFS

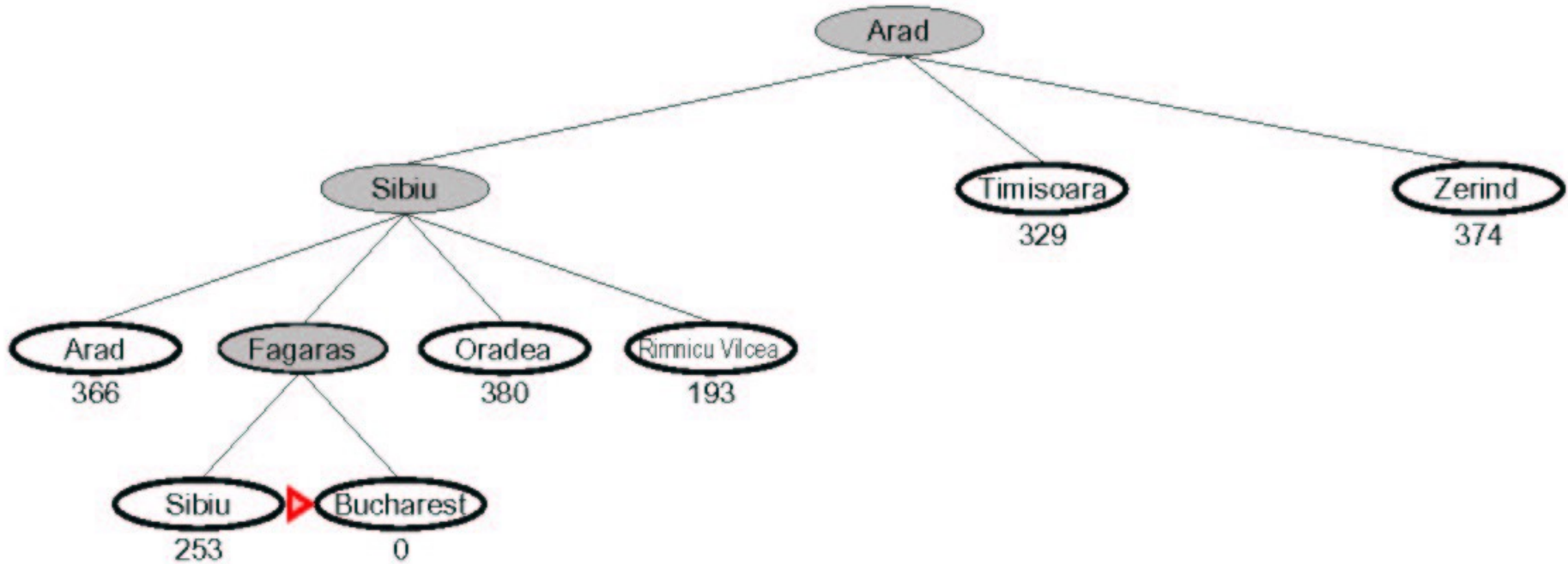
- Χρησιμοποιώντας ως ευρετική συνάρτηση την ευθεία ευκλείδεια απόσταση ως το στόχο (εδώ το Βουκουρέστι).



Αναζήτηση με BestFS



Αναζήτηση με BestFS

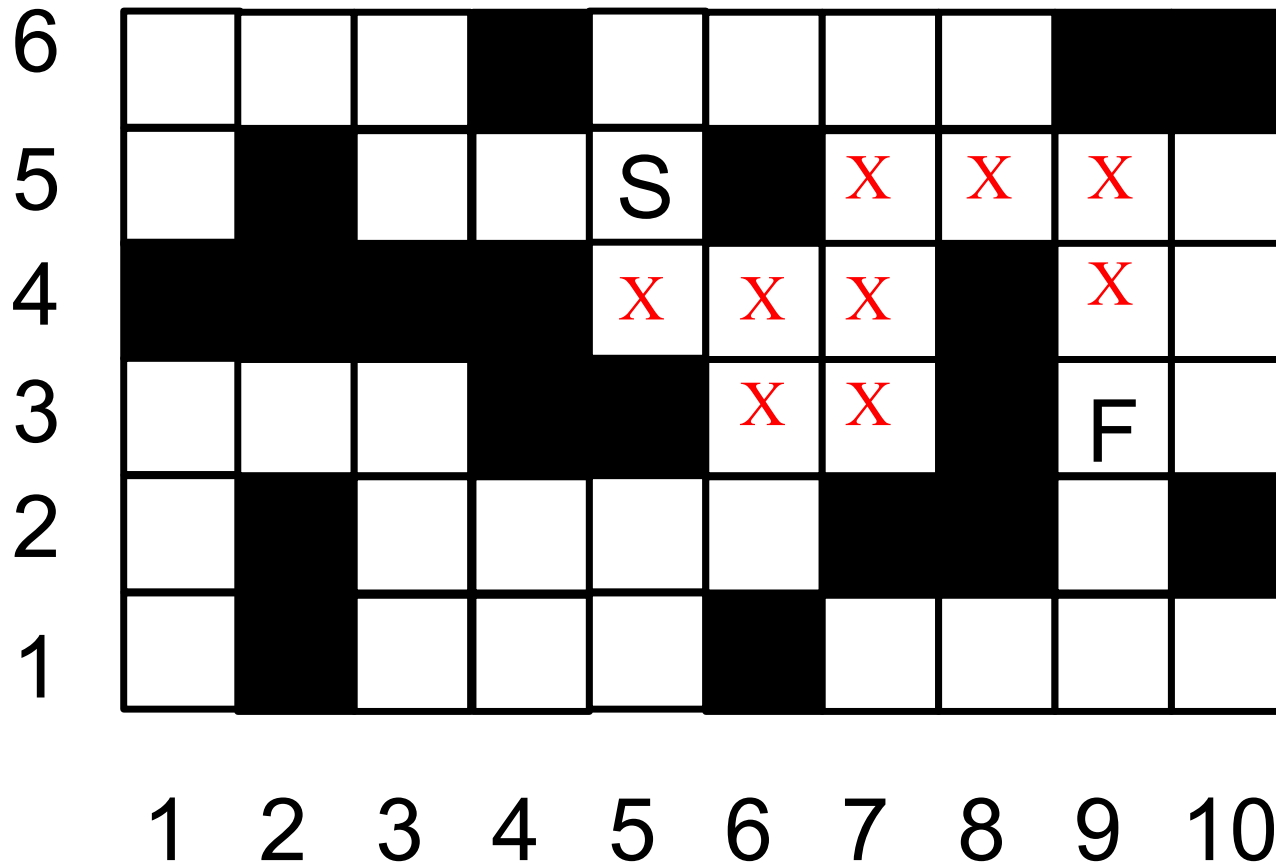


Δεν βρήκε το βέλτιστο μονοπάτι!

Χαρακτηριστικά BestFS (χωρίς Κ.Σ.)

- **Μη πλήρης:** μπορεί να παγιδευθεί σε άπειρα κλαδιά.
 - Ακόμα και σε πεπερασμένους χώρους καταστάσεων, μπορεί να παγιδευθεί σε παλινδρομήσεις (π.χ. προσπάθεια μετάβασης από Iasi σε Fagaras, παλινδρόμηση μεταξύ Neamt και Iasi).
- **Μη βέλτιστος:**
 - Εμπιστεύεται την h , που μπορεί να κάνει λάθος.
 - Και δεν λαμβάνει υπόψη του το κόστος μέχρι τα παιδιά που αξιολογεί η h (πόσο έχει κοστίσει ήδη το μονοπάτι ως εκεί).
- Πολυπλοκότητα **χρόνου**: $O(b^m)$. Όπως στον DFS.
- Πολυπλοκότητα **χώρου**: $O(b^m)$.
 - Αντίθετα από τον DFS, το μέτωπο δεν περιέχει σίγουρα μόνο παιδιά προγόνων ενός κόμβου και τη ρίζα.
- Όμως στην πράξη μια **καλή $h(n)$ μπορεί να μειώσει** πολύ το χρόνο και χώρο που θα χρειαστούμε.
 - Με **ιδανική ευρετική**, που θα μας καθοδηγεί κατά μήκος του βέλτιστου μονοπατιού, θα χρειαστούμε $O(bd)$ χρόνο και χώρο.

Αναζήτηση BestFS με κλειστό σύνολο



Αναζήτηση BestFS με κλειστό σύνολο

Μέτωπο	Κ.Σ.	Κατ/ση	Παιδιά
5-5	{}	5-5	5-4:5, 5-6:7, 4-5:7
5-4:5, 5-6:7, 4-5:7	{5-5}	5-4	5-5:6, 6-4:4
6-4:4, 5-5:6, 5-6:7, 4-5:7	+5-4	6-4	5-4:5, 6-3:3, 7-4:3
6-3:3, 7-4:3, 5-4:5, 5-5:6, ...	+6-4	6-3	6-4:4, 6-2:4, 7-3:2
7-3:2, 7-4:3, 6-2:4, 6-4:4, ...	+6-3	7-3	6-3:3, 7-4:3
6-3:3, 7-4:3, 6-2:4, 6-4:4, ...	+7-3	6-3	(βρόχος)
7-4:3, 6-2:4, 6-4:4, ...	(ίδιο)	7-4	7-5:4, 6-4:4, 7-3:2
7-3:2, 7-5:4, 6-4:4, 6-2:4, ...	+7-4	7-3	(βρόχος)
7-5:4, 6-4:4, 6-2:4, ...	(ίδιο)	7-5	7-4:3, 8-5:3, 7-6:5
8-5:3, 7-4:3, 7-5:4, 6-4:4, ...	+7-5	8-5	8-6:4, 7-5:4, 9-5:2
9-5:2, 7-4:3, 8-6:4, 7-5:4, ...	+8-5	9-5	8-5:3, 9-4:1, 10-5:3
9-4:1, 10-5:3, 8-5:3, 7-4:3, ...	+9-5	9-4	9-3:0, 9-5:2, 10-4:2
9-3:0, 9-5:2, 10-4:2, 10-5:3, ...	+9-4	9-3	(τελική)

Βιβλιογραφία

- Russel & Norvig (4^η έκδοση): ενότητες 3.4.3, 3.4.4, 3.4.6, 3.5.1.
 - Όσοι ενδιαφέρονται μπορούν να μελετήσουν προαιρετικά και την ενότητα 3.4.5.
- Βλαχάβας κ.ά.: υπόλοιπο κεφ. 3 (εκτός των ενοτήτων 3.4 και 3.5), εισαγωγή κεφ. 4, ενότητες 4.1, 4.2.
 - Όσοι ενδιαφέρονται μπορούν να διαβάσουν προαιρετικά (εκτός εξεταστέας ύλης) και τις ενότητες του κεφ. 3 που εξαιρέθηκαν.