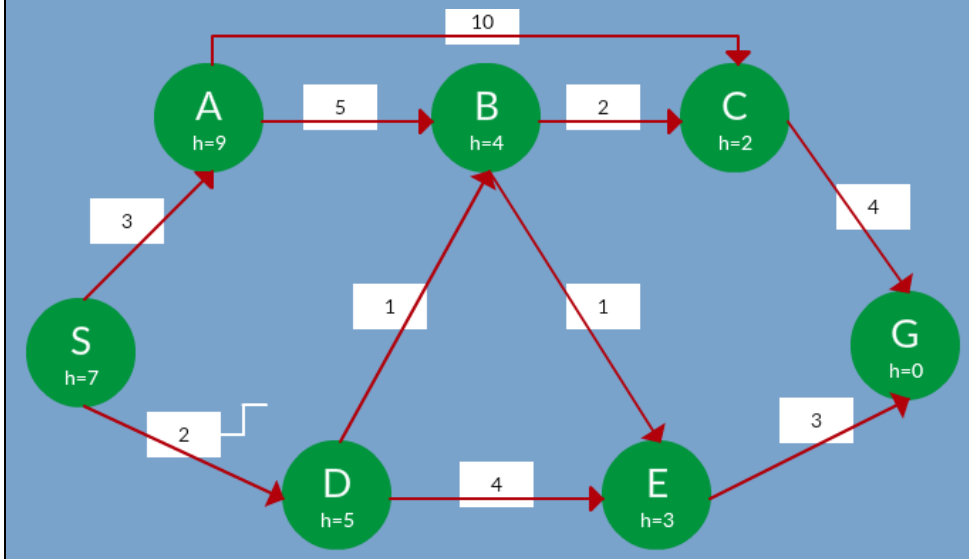
ΜΕΡΟΣ Α

Κεφάλαιο 4



Αλγόριθμοι Ευρετική ς Αναζήτησης

Αλγόριθμοι Ευρετικής Αναζήτησης

- Οι αλγόριθμοι τυφλής αναζήτησης προχωρούν σε βάθος ή πλάτος χωρίς ένδειξη για το αν το μονοπάτι που ακολουθούν οδηγεί σε κάποια τερματική κατάσταση.
- Σε πολλά προβλήματα ο χώρος αναζήτησης αυξάνεται ραγδαία (συνδυαστική έκρηξη).
  - Σε τέτοιους χώρους, η τυφλή αναζήτηση έχει τόσο μεγάλες απαιτήσεις σε χώρο και διαρκεί τόσο πολύ χρόνο που πρακτικά η λύση δεν βρίσκεται ποτέ
  - **Σκοπός**: Μείωση του αριθμού των καταστάσεων που επισκέπτεται ο αλγόριθμος.
  - **Απαιτείται**: Πληροφορία για αξιολόγηση των καταστάσεων που θα οδηγήσει γρηγορότερα στη λύση ή θα βοηθήσει στο κλάδεμα καταστάσεων που δεν οδηγούν πουθενά.
- Οι αλγόριθμοι που εκμεταλλεύονται τέτοιες πληροφορίες (κάποιον ευερετικό μηχανισμό) ονομάζονται Αλγόριθμοι Ευρετική ς Αναζήτησης.

Παράδειγμα

- Παράδειγμα Ευρετικής αναζήτησης είναι η συναρμολόγηση ενός puzzle.
  - Με τυφλή αναζήτηση θα έπρεπε να προσπαθήσουμε να το φτιάξουμε, χωρίς να δώσουμε καμία σημασία στο χρώμα ή το σχήμα των κομματιών.
  - Στην πραγματικότητα όμως, ενεργούμε εντελώς διαφορετικά. Χωρίζουμε τα κομμάτια σε κατηγορίες, π.χ. αυτά που ανήκουν στην περιφέρεια (μία άκρη τους είναι ευθεία τομή), σε αυτά που πιθανά ανήκουν στο χρώμα του ουρανού ή στη θάλασσα ή στα δένδρα κ.ο.κ.
  - Οι ενέργειες αυτές αντιστοιχούν στη χρήση ενός ευρετικού μηχανισμού που έχουμε αναπτύξει μέσω της εμπειρίας μας.
  - Ωστόσο, ποτέ δεν είμαστε σίγουροι ότι ο διαχωρισμός των κομματιών είναι και ο σωστός. Για παράδειγμα, υπάρχουν κομμάτια του puzzle που μπορεί να ανήκουν στη θάλασσα ή τον ουρανό.
- Ένα άλλο παράδειγμα είναι η εύρεση διαδρομής σε μία παραλιακή πόλη χτισμένη αμφιθεατρικά σε ένα λόφο, όπου το ζητούμενο είναι να βρεθεί κανείς δίπλα στη θάλασσα.
  - Στα σταυροδρόμια, ακολουθούμε τους κατηφορικούς δρόμους που πιθανά οδηγούν προς την κατεύθυνση της θάλασσας μιας και αυτή βρίσκεται στο χαμηλότερο υψόμετρο.
- Αν δεν υπήρχαν ευρετικοί μηχανισμοί, τότε τα προβλήματα αυτά θα λύνονταν πολύ δύσκολα, γιατί οι συνδυασμοί που πρέπει να γίνουν είναι πάρα πολλοί.
  - Θα ήταν σα να προσπαθεί κάποιος να συναρμολογήσει ένα puzzle από την ανάποδη πλευρά ή σα να ψάχνει την κεντρική πλατεία της πόλης με τα μάτια κλειστά.
- Ο ευρετικός μηχανισμός εξαρτάται από τη γνώση που έχουμε για το πρόβλημα.

Ευρετικός Μηχανισμός

Ευρετικός μηχανισμός (heuristic) είναι μία στρατηγική, βασισμένη στη γνώση για το συγκεκριμένο πρόβλημα, η οποία χρησιμοποιείται σα βοήθημα στη γρήγορη επίλυσή του.

- Ο ***ευρετικός μηχανισμός*** υλοποιείται με *ευρετική συνάρτηση* (*heuristic function*), που έχει πεδίο ορισμού το σύνολο των καταστάσεων ενός προβλήματος και πεδίο τιμών το σύνολο τιμών που αντιστοιχεί σε αυτές.
  - Τυπικά, μία ευρετική συνάρτηση Η ορίζεται ως $H:S \rightarrow R$, όπου S ο χώρος καταστάσεων και R το σύνολο των πραγματικών αριθμών.
  - Σε όμοια προβλήματα χρησιμοποιούμε παρόμοιους ευρετικούς μηχανισμούς.
  - Για το ίδιο πρόβλημα μπορεί να οριστούν πολλές ευρετικές συναρτήσεις.
  - Ανάλογα με τη λεπτομέρεια στην οποία υπεισέρχονται, οι ευρετικές συναρτήσεις χωρίζονται σε συναρτήσεις λεπτής υφής ή λεπτομερειακές (*fine grain*) και συναρτήσεις αδρής υφής ή μη-λεπτομερειακές (*coarse grain*).
    - ✓ Οι πρώτες δίνουν καλύτερη εκτίμηση, αλλά είναι πιο χρονοβόρες στον υπολογισμό τους.
- ***Ευρετική τιμή*** (*heuristic value*) είναι η τιμή της Ευρετικής συνάρτησης και εκφράζει το πόσο κοντά βρίσκεται μία κατάσταση σε μία τελική.
  - Η ευρετική τιμή δεν είναι η πραγματική τιμή της απόστασης από μία τερματική κατάσταση, αλλά μία εκτίμηση (estimate*)* που πολλές φορές μπορεί να είναι και λανθασμένη.

Ευριστικές Συναρτήσεις σε Μικρά Προβλήματα (1/3)

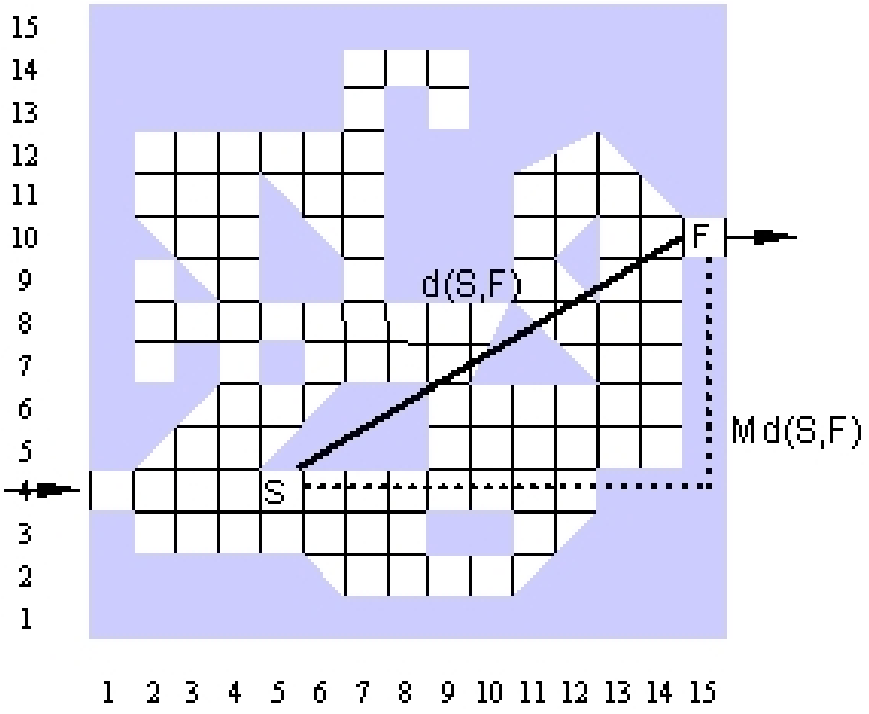
Ευρετικός μηχανισμός και συναρτήσεις σε λαβύρινθο

- Ευκλείδεια απόσταση (Euclidian distance):

$$d(\mathrm{S,F}) = \sqrt{(X_S - X_F)^2 + (Y_S - Y_F)^2}$$

- Απόσταση Manhattan[1] (Manhattan distance):

$$Md(S,F) = |X_S - X_F| + |Y_S - Y_F|$$



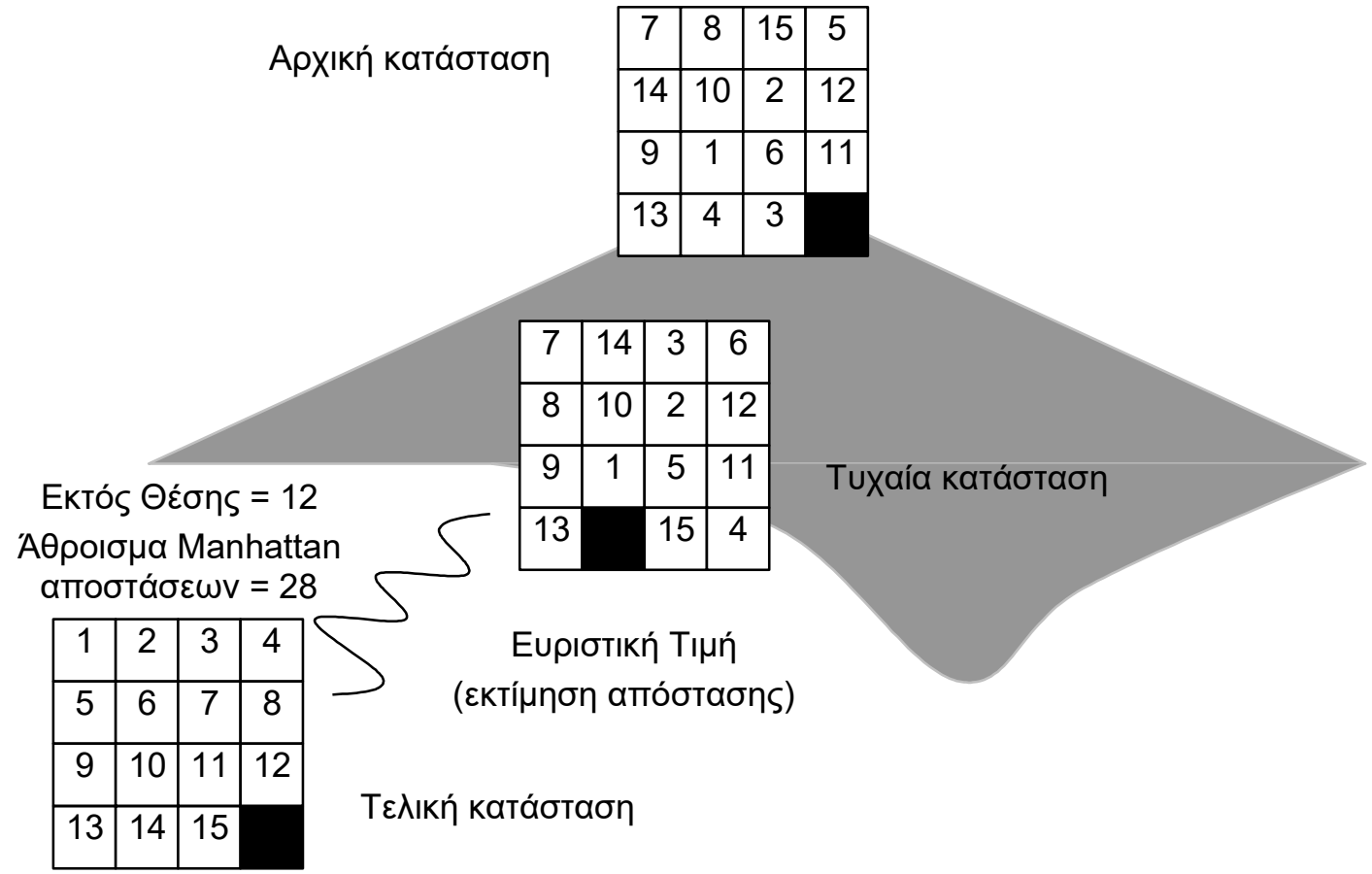
$$d(\mathrm{S,F}) = \sqrt{(5\text{-}15)^2 + (4\text{-}10)^2} = \sqrt{(100 + 36)} = 11{,}6$$

Md(S,F)=|5-15| + |4-10|=10+6=16.

[1] Ονομάζεται έτσι επειδή στο Manhattan της Νέας Υόρκης δεν μπορεί κανείς να κινηθεί διαγώνια γιατί η χωροθέτηση των κτιρίων επιτρέπει μόνον κάθετες και οριζόντιες λεωφόρους.

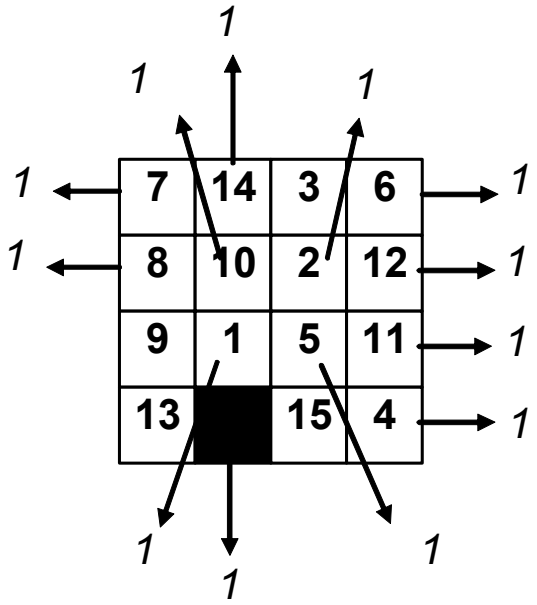
Ευρετικός μηχανισμός και συναρτήσεις στο N-Puzzle (1/2)

- Η διαφορά μίας κατάστασης από την τελική, η οποία αποτελεί εκτίμηση του αριθμού των κινήσεων που απαιτούνται για να επιτευχθεί η τελική κατάσταση
  - Πόσα πλακίδια βρίσκονται εκτός θέσης.
  - Το άθροισμα των αποστάσεων Manhattan κάθε πλακιδίου από την τελική του θέση.

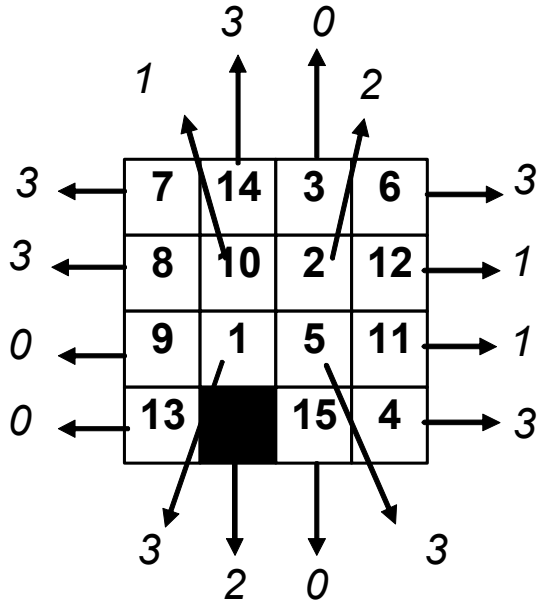


Ευρετικός μηχανισμός και συναρτήσεις στο N-Puzzle (2/2)

- Αναλυτικός υπολογισμός Ευρετική ς τιμής για μία τυχαία κατάσταση του 15-puzzle.



Εκτός θέσης = 12



Άθροισμα αποστάσεων Manhattan = 28

Ευρετικός μηχανισμός και συναρτήσεις στο TSP

- Η κοντινότερη πόλη έχει περισσότερες πιθανότητες να οδηγήσει σε μία συνολικά καλή λύση.
  - Το σκεπτικό είναι ότι πιθανά το μονοπάτι που ακολουθείται θα δώσει καλή λύση και (αν χρησιμοποιηθούν B&B τύπου αλγόριθμοι) το κλάδεμα που θα ακολουθήσει θα μειώσει το δένδρο αναζήτησης όσο το δυνατόν περισσότερο.

Αναζήτηση Πρώτα στο Καλύτερο

- Ο αλγόριθμος *αναζήτηση πρώτα στο καλύτερο* (*Best-First - BestFS*) επεκτείνει τις καταστάσεις από το μέτωπο αναζήτησης που θεωρεί καλύτερες, δηλαδή που εκτιμά ότι βρίσκονται πιο κοντά σε μια τελική κατάσταση.
- Κρατά όλες τις καταστάσεις στο μέτωπο αναζήτησης και μπορεί να επιστρέψει σε μία προγενέστερη κατάσταση αν το τρέχον μονοπάτι που ακολουθεί βρίσκεται σε χειρότερο δρόμο.

Ο αλγόριθμος BestFS

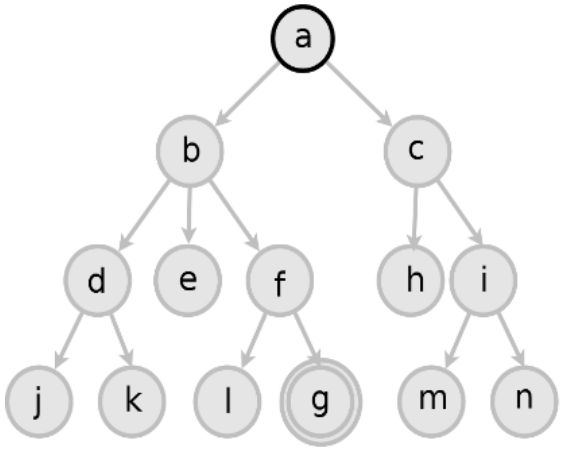
1. **Βάλε την αρχική κατάσταση στο μέτωπο αναζήτησης.**
2. **Αν το μέτωπο αναζήτησης είναι κενό τότε σταμάτησε.**
3. **Πάρε την πρώτη σε σειρά κατάσταση από το μέτωπο αναζήτησης.**
4. **Αν η κατάσταση είναι μέλος του κλειστού συνόλου τότε πήγαινε στο 2.**
5. **Αν η κατάσταση είναι μία τελική τότε ανέφερε τη λύση και σταμάτα.**
6. **Εφάρμοσε τους τελεστές μεταφοράς για να παράγεις τις καταστάσεις-παιδιά.**
7. **Εφάρμοσε την ευρετική συνάρτηση σε κάθε παιδί.**
8. **Βάλε τις καταστάσεις-παιδιά στο μέτωπο αναζήτησης.**
9. **Αναδιάταξε το μέτωπο αναζήτησης, έτσι ώστε η κατάσταση με την καλύτερη ευρετική τιμή να είναι πρώτη.**
10. **Βάλε τη κατάσταση-γονέα στο κλειστό σύνολο.**
11. **Πήγαινε στο βήμα 2.**

Ο αλγόριθμος BestFS (Ψευδοκώδικας)

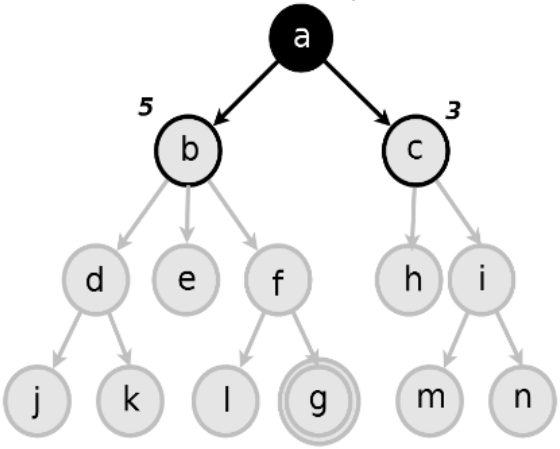
```
algorithm bestfs(InitialState, FinalStates)
begin
  Closed←∅;
  EvaluatedInitialState←Heuristic(<InitialState>)
  Frontier←<EvaluatedInitialState>;
  CurrentState←best(Frontier);
  while CurrentState ∉ FinalStates do
    Frontier←delete(CurrentState,Frontier);
    if CurrentState ∉ ClosedSet then
    begin
      Children ←Expand(CurrentState);
      EvaluatedChildren←Heuristic(Children);
      Frontier←Frontier ^ EvaluatedChildren;
      Closed←Closed∪{CurrentState};
    end;
    if Frontier= ∅ then return fail;
    CurrentState←best(Frontier);
  endwhile;
  return success;
end.
```

Αναζήτηση Πρώτα στο Καλύτερο σε χώρο αναζήτησης με αρχική κατάσταση a και τελική g

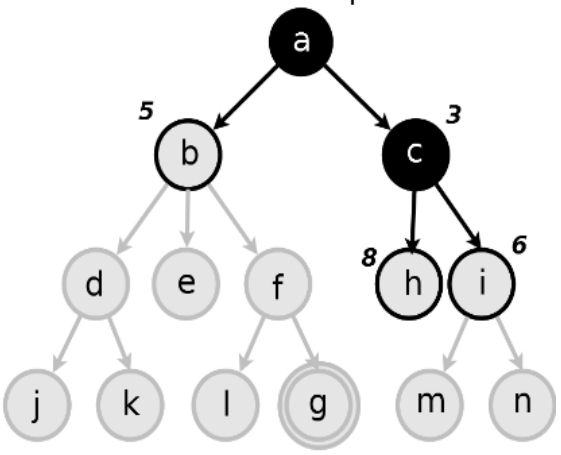
Αρχική Κατάσταση



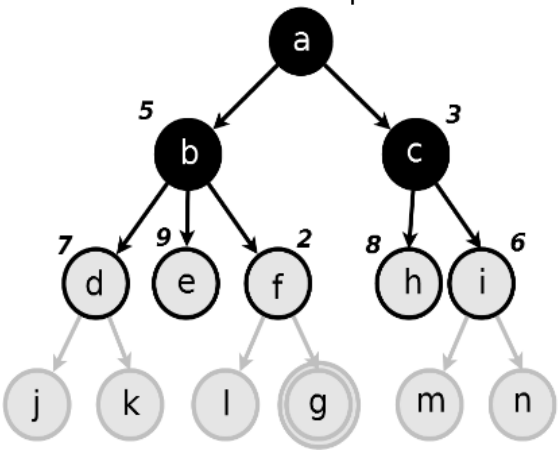
Επέκταση a



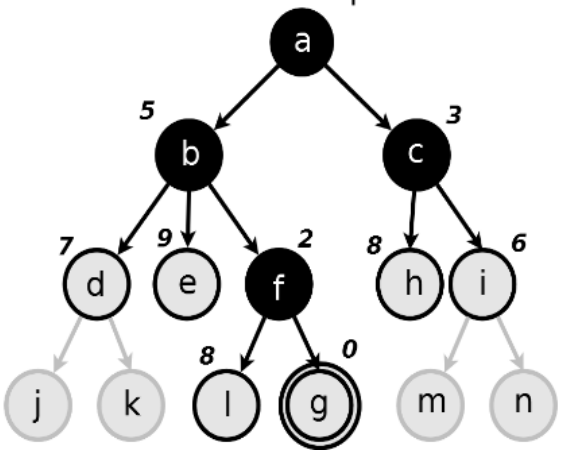
Επέκταση c



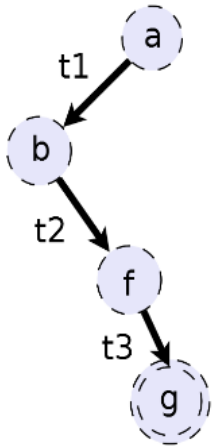
Επέκταση b



Επέκταση f



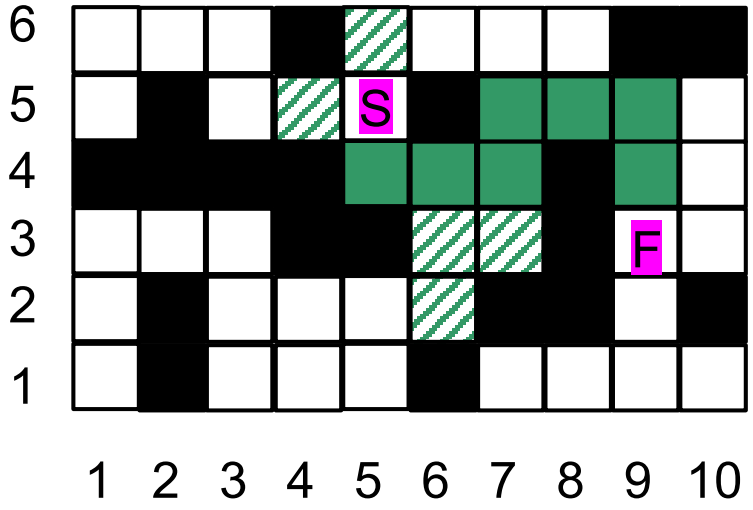
Λύση



Ο αλγόριθμος BestFS

Σχόλια

- Πλεονεκτήματα:
  - Προσπαθεί να δώσει μια γρήγορη λύση σε κάποιο πρόβλημα
    - ✓ Το αν τα καταφέρει ή όχι εξαρτάται πολύ από τον ευερετικό μηχανισμό
  - Είναι πλήρης
- Μειονεκτήματα:
  - Το μέτωπο αναζήτησης μεγαλώνει με υψηλό ρυθμό και μαζί του ο χώρος που χρειάζεται για την αποθήκευσή του, καθώς και ο χρόνος για την επεξεργασία των στοιχείων του.
  - Δεν εγγυάται ότι η λύση που θα βρεθεί είναι η βέλτιστη.



Ο αλγόριθμος BestFS: το πρόβλημα του λαβύρινθου

Μέτωπο Αναζήτησης	Κλειστό Σύνολο	Κατάσταση	Παιδιά
<5-5>	<>	5-5	$5\text{-}4^{5},5\text{-}6^{7},4\text{-}5^{7}$
$<5\text{-}4^{5},5\text{-}6^{7},4\text{-}5^{7}>$	<5-5>	5-4	$5\text{-}5^{6},6\text{-}4^{4}$
$<6\text{-}4^{4},5\text{-}5^{6},5\text{-}6^{7},4\text{-}5^{7}>$	<5-5,5-4>	6-4	$5\text{-}4^{7},6\text{-}3^{3},7\text{-}4^{3}$
$<6\text{-}3^{3},7\text{-}4^{3},5\text{-}5^{6},5\text{-}6^{7},\ldots>$	<5-5,5-4,6-4>	6-3	$6\text{-}4^{4},6\text{-}2^{3},7\text{-}3^{2}$
$<7\text{-}3^{2},6\text{-}2^{3},7\text{-}4^{3},6\text{-}4^{4},5\text{-}5^{6},\ldots>$	<5-5,5-4,…>	7-3	$6\text{-}3^{3},6\text{-}4^{4}$
$<6\text{-}3^{3},6\text{-}2^{3},7\text{-}4^{3},6\text{-}4^{4},5\text{-}5^{6},\ldots>$	<…,6-3,…>	6-3	Βρόχος
$<6\text{-}2^{3},7\text{-}4^{3},6\text{-}4^{4},5\text{-}5^{6},5\text{-}6^{7},\ldots>$	<…>	6-2	$5\text{-}2^{5},6\text{-}3^{3}$
$<7\text{-}4^{3},6\text{-}4^{4},5\text{-}2^{5},\ldots>$	<…>	7-4	$7\text{-}5^{4},6\text{-}4^{4},7\text{-}3^{2}$
$<7\text{-}3^{2},7\text{-}5^{4},6\text{-}4^{4},5\text{-}2^{5},\ldots>$	<..,7-3,..>	7-3	Βρόχος
$<4\text{-}5^{4},6\text{-}4^{4},5\text{-}2^{5},\ldots>$	<…>	7-5	$7\text{-}4^{3},8\text{-}5^{3},7\text{-}6^{5}$
$<8\text{-}5^{3},7\text{-}4^{3},6\text{-}4^{4},\ldots>$	<…>	8-5	$8\text{-}6^{4},7\text{-}5^{4},9\text{-}5^{2}$
$<9\text{-}5^{2},7\text{-}4^{3},6\text{-}4^{4},8\text{-}6^{4},\ldots>$	<…>	9-5	$8\text{-}5^{3},9\text{-}4^{1}$
$<9\text{-}4^{1},8\text{-}5^{3},7\text{-}4^{3},\ldots>$	<…>	9-4	$9\text{-}3^{0},9\text{-}5^{2},10\text{-}4^{2}$
$<9\text{-}3^{0},9\text{-}5^{2},10\text{-}4^{2},\ldots>$	<…>	9-3	ΤΕΛΙΚΗ ΚΑΤΑΣΤΑΣΗ
		ΤΕΛΟΣ	

- Η λύση στο παραπάνω πρόβλημα είναι η διαδρομή που ορίζεται από τη σειρά των θέσεων: 5-5→5-4→6-4→7-4→7-5→8-5→9-5→9-4→9-3

Αναζήτηση με Αναρρίχηση Λόφων

Hill-Climbing Search – HC

- Επεκτείνει την κατάσταση που εκτιμά ότι είναι η καλύτερη και τις άλλες τις κλαδεύει
- Δηλαδή στο μέτωπο αναζήτησης υπάρχει μόνο μία κατάσταση
- Ο αλγόριθμος μοιάζει πολύ με τον DFS.

Ο αλγόριθμος HC

1. **Η αρχική κατάσταση είναι η τρέχουσα κατάσταση.**
2. **Αν η κατάσταση είναι μία τελική τότε ανέφερε τη λύση και σταμάτησε.**
3. **Εφάρμοσε τους τελεστές μετάβασης για να βρεις τις καταστάσεις-παιδιά.**
4. **Βρες την καλύτερη κατάσταση σύμφωνα με την ευρετική συνάρτηση.**
5. **Η καλύτερη κατάσταση γίνεται η τρέχουσα κατάσταση.**
6. **Πήγαινε στο βήμα 2.**

Ο αλγόριθμος HC (Ψευδοκώδικας)

```
algorithm hc(InitialState, FinalState)
  begin
    CurrentState←InitialState;
    while CurrentState≠FinalState do
        Children←Expand(CurrentState);
        if Children=∅ then return failure;
        EvaluatedChildren←Heuristic(Children);
        bestChild←best(EvaluatedChildren);
        if hValue(CurrentState) ≥ hValue(bestChild)
          then return failure;
          else CurrentState←bestChild;
        endif;
    endwhile;
    return success;
  end.
```

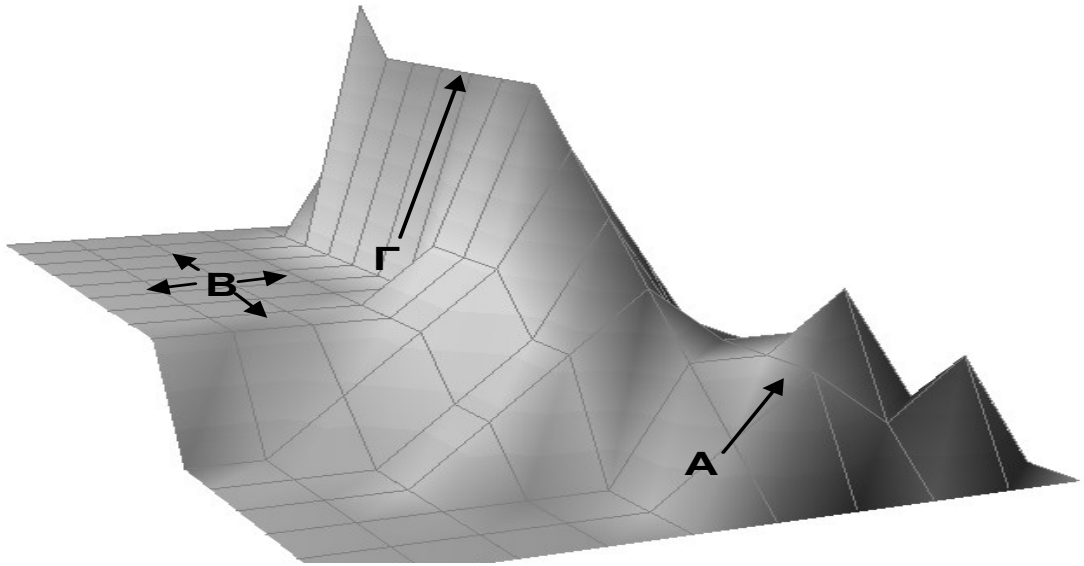
Ο αλγόριθμος HC

Σχόλια (1/2)

- Ο HC χρησιμοποιείται σε προβλήματα όπου πρέπει να βρεθεί μία λύση πολύ γρήγορα, έστω και αν αυτή δεν είναι η καλύτερη, παίρνοντας όμως και το ρίσκο να μη βρεθεί καμία λύση, έστω και αν τέτοια υπάρχει.
- Πλεονεκτήματα:
  - Πολύ αποδοτικός και σε χρόνο και σε μνήμη
- Μειονεκτήματα:
  - Είναι μη-πλήρης

❖ Βασικά προβλήματα του HC:

- ❑ Ας υποτεθεί ότι ο χώρος αναζήτησης είναι μια επιφάνεια με λόφους, πεδιάδες και χαράδρες και αναζητείται η ψηλότερη κορυφή.
- ❑ Η αναζήτηση γίνεται μέσα στην ομίχλη και συνεπώς δεν υπάρχει πραγματική πληροφορία για το πού βρίσκεται η ψηλότερη κορυφή.
  - ✓ Πρόποδες (foothill). Κάποιες χαμηλότερες κορυφές προσελκύουν το ενδιαφέρον της αναζήτησης. Έτσι η λύση που βρίσκεται είναι η τοπικά καλύτερη (local maximum) και όχι η συνολικά καλύτερη.
  - ✓ Οροπέδιο (plateau). Υπάρχει η περίπτωση όλα τα γύρω σημεία να έχουν το ίδιο ακριβώς ύψος. Η επιλογή ενός από τα ισοϋψή σημεία γίνεται τυχαία και η περιπλάνηση γύρω από την ίδια περιοχή είναι αναπόφευκτη.
  - ✓ Κορυφογραμμή (ridges). Υπάρχει περίπτωση η ευρετική τιμή να εναλλάσσεται μεταξύ δύο τιμών, δηλαδή των σημείων που βρίσκονται εκατέρωθεν της κορυφογραμμής, χωρίς όμως να υπάρχει ουσιαστική πρόοδος.



- ❑ Για να ξεπεραστούν τα προβλήματα αυτά πρέπει η ευρετική συνάρτηση να καλύπτει μεγαλύτερο εύρος, δηλαδή ουσιαστικά η ορατότητά της να είναι μεγαλύτερη.

Βελτιώσεις:

- Ακτινωτή Αναζήτηση
- Αναζήτηση με απαγορευμένες καταστάσεις (Tabu Search - TS)
- Εξαναγκασμένη αναρρίχηση λόφου (Enforced Hill-Climbing - EHC)
- Προσομοιωμένη ανόπτηση (Simulated Annealing - SA)

Ακτινωτή Αναζήτηση - Beam Search - BS

Στον *αλγόριθμο ακτινωτής αναζήτησης* δεν κλαδεύονται όλες οι υπόλοιπες καταστάσεις όπως στον HC, αλλά ένας σταθερός αριθμός Κ από τις καλύτερες από αυτές κρατείται στο μέτωπο αναζήτησης.

- Είναι μία μέση προσέγγιση ανάμεσα στον HC και τον BestFS
  - Αν υπάρχουν περισσότερες από μία υπολογιστικές μονάδες τότε η επέκταση των καταστάσεων μπορεί να γίνει παράλληλα

Ο αλγόριθμος BS

1. Βάλε την αρχική κατάσταση στο μέτωπο αναζήτησης.
2. Αν το μέτωπο αναζήτησης είναι κενό τότε σταμάτησε.
3. Πάρε την πρώτη σε σειρά κατάσταση από το μέτωπο αναζήτησης.
4. Αν η κατάσταση είναι μέλος του κλειστού συνόλου τότε πήγαινε στο 2.
5. Αν η κατάσταση είναι μία τελική τότε ανέφερε τη λύση και σταμάτα.
6. Εφάρμοσε τους τελεστές μεταφοράς για να παράγεις τις καταστάσεις-παιδιά.
7. Εφάρμοσε την ευρετική συνάρτηση σε κάθε παιδί.
8. Βάλε τις καταστάσεις-παιδιά στο μέτωπο αναζήτησης.
9. Αναδιάταξε το μέτωπο αναζήτησης, έτσι ώστε η κατάσταση με την καλύτερη ευρετική τιμή να είναι πρώτη.
10. Από όλο το μέτωπο της αναζήτησης κράτα μόνο τις Κ καλύτερες καταστάσεις.
11. Βάλε τη κατάσταση-γονέα στο κλειστό σύνολο.
12. Πήγαινε στο βήμα 2.

Αναζήτηση με Απαγορευμένες Καταστάσεις (Tabu Search – TS)

- Παραλλαγή της αναζήτησης HC
  - Από τις γειτονικές καταστάσεις μιας κατάστασης S σε κάθε βήμα επιλέγεται η S', ακόμη και αν αυτή αξιολογείται από την ευρετική συνάρτηση ως χειρότερη από την S
    - ✓ Αυτή είναι και η διαφορά από τον HC
  - Παράλληλα, υπάρχει μια λίστα με καταστάσεις τις οποίες ο αλγόριθμος TS έχει επισκεφθεί πρόσφατα ή δεν πρέπει να επισκεφτεί γιατί υπάρχουν σε αυτές κάποια χαρακτηριστικά (περιορισμοί) που τις κάνουν ανεπιθύμητες.
    - ✓ Αυτή είναι η λεγόμενη λίστα απαγορευμένων καταστάσεων (*tabu list*)
  - Η λίστα *tabu* λαμβάνεται υπόψη στην επιλογή της επόμενης κατάστασης, αφού ποτέ δεν επιλέγεται μια κατάσταση που είναι μέλος αυτής της λίστας
  - Μοιάζει με το κλειστό σύνολο. Η διαφορά είναι ότι έχει συγκεκριμένο μέγεθος
    - ✓ Η εισαγωγή νέων καταστάσεων έχει ως αποτέλεσμα τη διαγραφή προηγούμενων μελών

Εξαναγκασμένη Αναρρίχηση Λόφου

Enforced Hill-Climbing - EHC

- Παραλλαγή του αλγορίθμου αναρρίχησης λόφου (HC)
- Συνδυασμός της απλής αναρρίχησης λόφου και του αλγορίθμου BFS.
- Ο EHC σε κάθε βήμα, όπως και ο HC, κάνει προσπάθεια να βρει μια νέα κατάσταση που να θεωρείται καλύτερη, σύμφωνα με τη συνάρτηση αξιολόγησης.
- Εάν κάτι τέτοιο δεν είναι δυνατό, ο EHC εκτελεί μια αναζήτηση κατά πλάτος από την τρέχουσα κατάσταση, μέχρι να βρεθεί μια καλύτερη κατάσταση.
  - Μόλις βρεθεί, η αναζήτηση συνεχίζει από αυτήν
- Το πλεονέκτημα της εξαναγκασμένης αναζήτησης είναι ότι μπορεί να ξεφεύγει από σχετικά μικρά οροπέδια.

Προσομοιωμένη Ανόπτηση

Simulated Annealing - SA

- Παραλλαγή του HC
- Δίνει μια πιθανότητα μετάβασης σε χειρότερες καταστάσεις, αφήνοντας έτσι ένα ενδεχόμενο να ξεφύγει η αναζήτηση από τοπικά ακρότατα.
- Η μέθοδος αυτή προέρχεται από τις φυσικές διαδικασίες δημιουργίας κρυσταλλικών δομών στη φύση.
  - Πιο συγκεκριμένα, κατά την αναζήτηση SA επιλέγεται στην τύχη μια κατάσταση S' από τις επόμενες καταστάσεις της τρέχουσας κατάστασης S.
  - Εάν η κατάσταση αυτή αξιολογηθεί ως καλύτερη από την προηγούμενη, δηλαδή εάν $\Delta=h(S')-h(S)\leq 0$, τότε η αναζήτηση SA συνεχίζει από την S'.
  - Εάν όμως $\Delta=h(S')-h(S)>0$, τότε η μετάβαση στην S' πραγματοποιείται με πιθανότητα $e^{-\Delta/t}$, όπου t μια παράμετρος που ονομάζεται θερμοκρασία, κατά αναλογία των φυσικών διεργασιών που αναφέρθηκαν παραπάνω.
  - Για $t\rightarrow 0$, ο αλγόριθμος τείνει να λειτουργεί ως ο HC.
  - Όσο μεγαλύτερη είναι η τιμή Δ, αλλά και όσο μικρότερη είναι η τιμή της θερμοκρασίας, τόσο μικρότερη είναι η πιθανότητα να πραγματοποιηθεί η μετάβαση από την S στην S'.

Ο Αλγόριθμος Άλφα-Άστρο (A*)

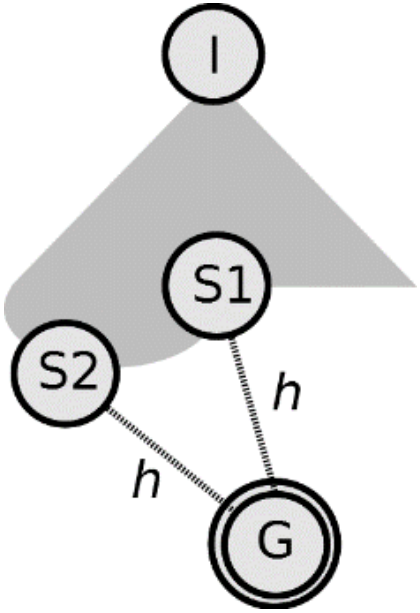
Ο *αλγόριθμος Άλφα Άστρο (A star)* είναι κατά βάση BestFS, αλλά με ευρετική συνάρτηση:

F(S) = g(S) + h(S)

η g(S) δίνει την απόσταση της S από την αρχική κατάσταση, η οποία είναι πραγματική και γνωστή, και

η h(S) δίνει την εκτίμηση της απόστασης της S από την τελική κατάσταση μέσω μιας Ευρετικής συνάρτησης, όπως ακριβώς στον BestFS.

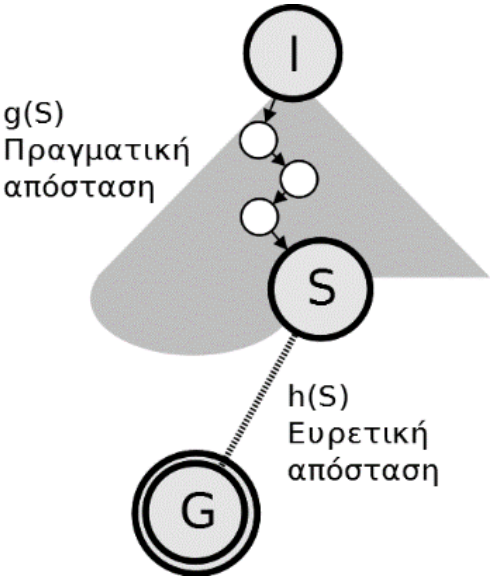
- Ο BestFS δεν εγγυάται ότι η λύση που θα βρει σε ένα πρόβλημα θα είναι η καλύτερη.
- Αυτό προκύπτει από το ότι ανάμεσα σε δύο ισοδύναμες καταστάσεις (π.χ. S1 και S2 που έχουν ίδια ευρετική τιμή) θα διαλέξει μία τυχαία για να συνεχίσει την αναζήτηση.
- Αν υποθέσουμε ότι εν δυνάμει και οι δύο οδηγούσαν σε λύση αλλά η S2 ήταν πιο βαθιά στο δένδρο από την S1, τότε το μονοπάτι της λύσης είναι πιο μεγάλο σε μήκος από αυτό που θα περνούσε από την S1.
- Αν όμως ληφθεί υπόψη και το βάθος που βρίσκεται η κάθε κατάσταση, τότε ο αλγόριθμος θα είχε διαλέξει την S1.



Ο Αλγόριθμος Άλφα-Άστρο (A*)

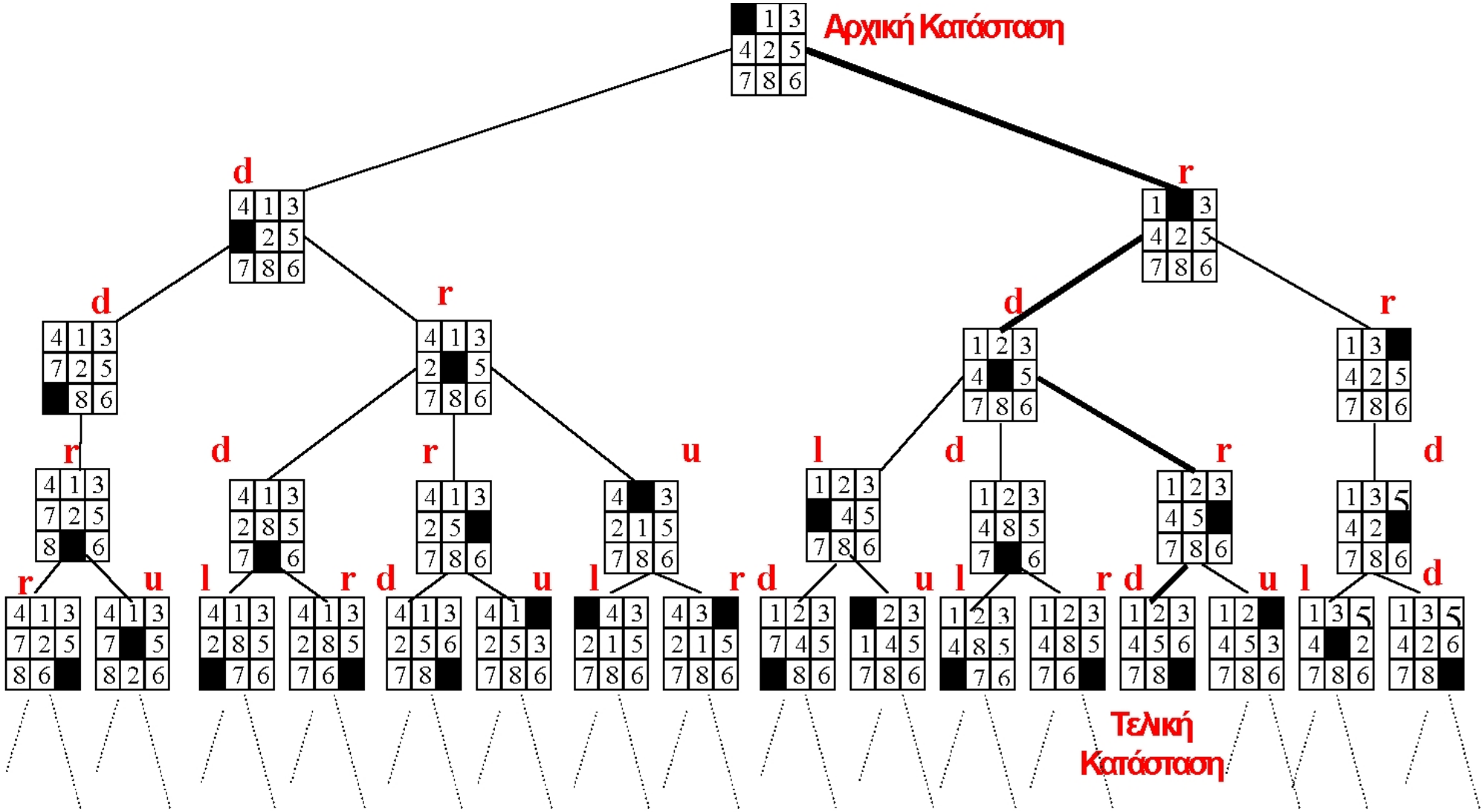
Σχόλια

- Ενώ στον BestFS η ευρετική συνάρτηση δίνει μόνο την εκτίμηση της απόστασης μιας κατάστασης από την τελική, στον A* η τιμή αυτή προστίθεται στην απόσταση που έχει ήδη διανυθεί από την αρχική μέχρι την τρέχουσα κατάσταση,
- Δηλαδή: F(S) = g(S) + h(S)
  - Αν για κάθε κατάσταση η τιμή h(S) είναι μικρότερη ή το πολύ ίση με την πραγματική απόσταση της S από την τελική κατάσταση, τότε ο A* βρίσκει πάντα τη βέλτιστη λύση.
  - Στην περίπτωση αυτή, ο ευρετικός μηχανισμός ονομάζεται *αποδεκτός* (*admissible*) και ικανοποιεί το *κριτήριο αποδοχής* (*admissibility criterion*).
- Βελτιώσεις:
  - A* με επαναληπτική εκβάθυνση (Iterative Deepening A* - IDA)

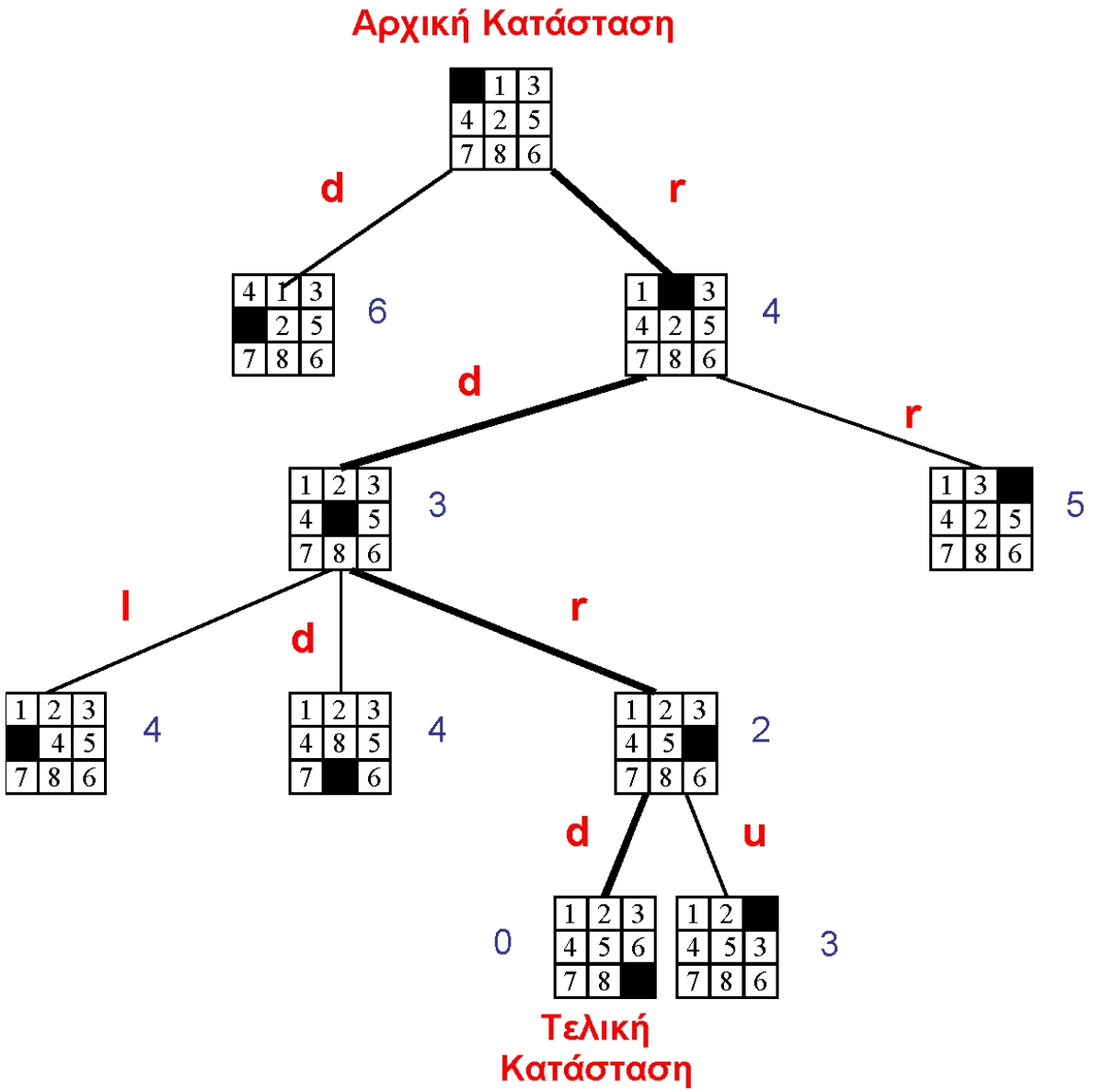


Εφαρμογή των Αλγορίθμων Ευρετική ς Αναζήτησης

Χώρος Αναζήτησης στο 8-puzzle

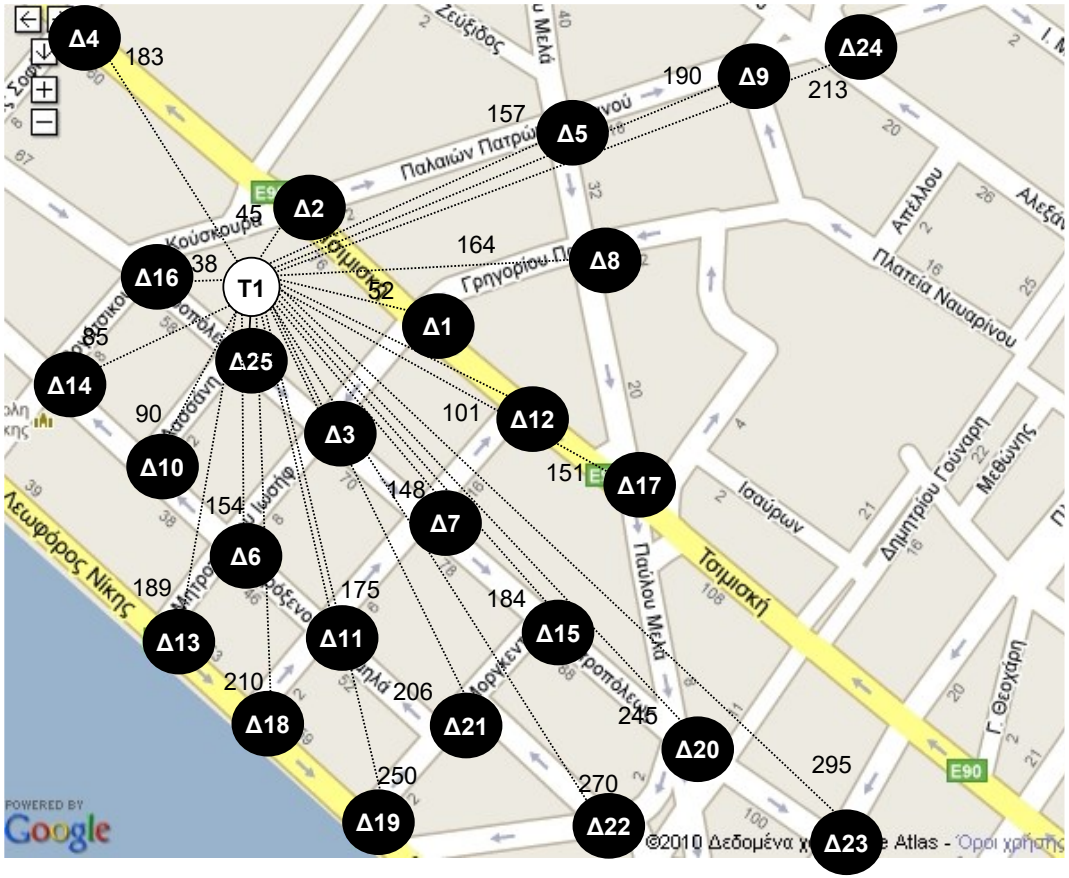


Εφαρμογή αλγορίθμου BestFS στο 8-puzzle



Εύρεση διαδρομής

- Τελική κατάσταση Τ1. Ευρετική συνάρτηση: Ευκλείδεια απόσταση
- Στον BestFS από το μέτωπο της αναζήτησης επεκτείνεται η κατάσταση με την μικρότερη τιμή
- Στον Α* σε κάθε διασταύρωση επιλέγεται (επεκτείνεται) η κατάσταση με το μικρότερο άθροισμα (εκτιμώμενη απόσταση από την τελική κατάσταση + απόσταση που έχει ήδη διανυθεί από την αρχική)



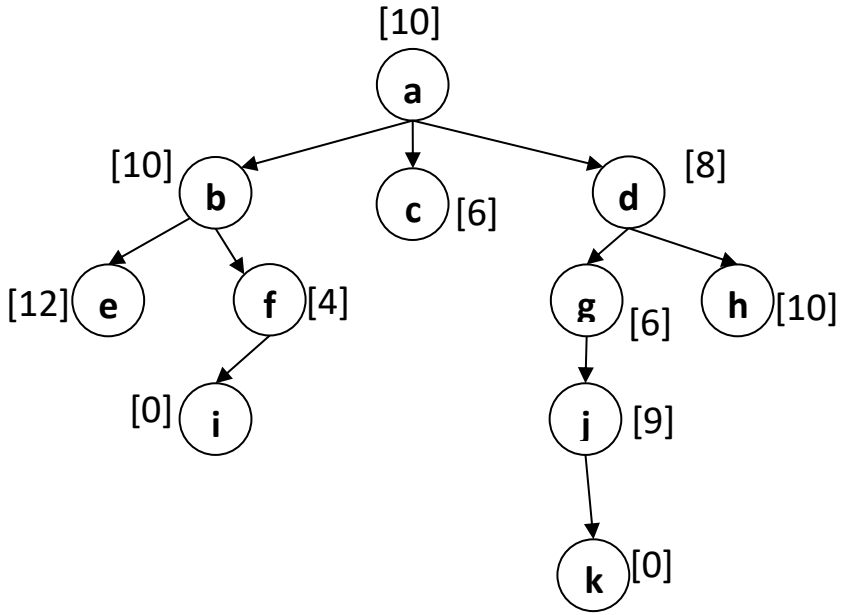
Ερωτήσεις Κεφαλαίου 4

1. Γιατί δεν είναι πλήρης ο αλγόριθμος Αναρρίχησης Λόφων;
2. Πλεονεκτήματα, μειονεκτήματα, BestFS, HC, A* (ένα από όλα)
3. Ποια είναι η κύρια διαφορά μεταξύ των αλγορίθμων Πρώτα στο Καλύτερο (BestFS) και Αναρρίχησης Λόφων (HC):
4. Σκοπός της Ευρετική ς αναζήτησης είναι να μειωθεί ……., δηλαδή ουσιαστικά να μειωθεί ……. που εξετάζει ένας αλγόριθμος.

Κυκλώστε το αντίστοιχο γράμμα Σ (ωστό) – Λ (άθος) στις επόμενες ερωτήσεις: (Προσοχή Σε αυτό το θέμα υπάρχει αρνητική βαθμολογία)		
Οι ευρετικοί αλγόριθμοι δεν ενδείκνυνται για προβλήματα με μεγάλους παράγοντες διακλάδωσης	Σ	Λ
Ο HC δεν έχει δυνατότητα οπισθοδρόμησης	Σ	Λ
Ο αλγόριθμος αναζήτησης πρώτα στο καλύτερο (Best-First - BestFS) δεν εγγυάται ότι η πρώτη λύση που θα βρεθεί είναι η βέλτιστη	Σ	Λ
Σκοπός της Ευρετική ς αναζήτησης είναι η μείωση του χώρου αναζήτησης	Σ	Λ
Ο Best First Search επιστρέφει πάντα την βέλτιστη λύση	Σ	Λ
Ο αλγόριθμος αναζήτησης A* εγγυάται ότι θα βρει τη βέλτιστη λύση ανεξάρτητα από την ευρετική συνάρτηση	Σ	Λ
Όταν η ευρετική συνάρτηση δίνει πάντα υποεκτιμήσεις της πραγματικής απόστασης, κανένας ευρετικός αλγόριθμος δεν εγγυάται την εύρεση βέλτιστης λύσης.	Σ	Λ
Ο αλγόριθμος αναζήτησης A* εγγυάται ότι η πρώτη λύση που θα βρεθεί είναι η βέλτιστη ανεξάρτητα από την ευρετική συνάρτηση	Σ	Λ
Ο HC με αποδεκτή ευρετική συνάρτηση επιστρέφει πάντα την βέλτιστη λύση	Σ	Λ
Ο αλγόριθμος αναζήτησης πρώτα στο καλύτερο (Best-First - BestFS) εγγυάται ότι η πρώτη λύση που θα βρεθεί είναι η βέλτιστη	Σ	Λ
Σκοπός της Ευρετική ς αναζήτησης είναι η μείωση του χώρου καταστάσεων	Σ	Λ
Ο αλγόριθμος BFS είναι πλήρης	Σ	Λ

Άσκηση 4.1

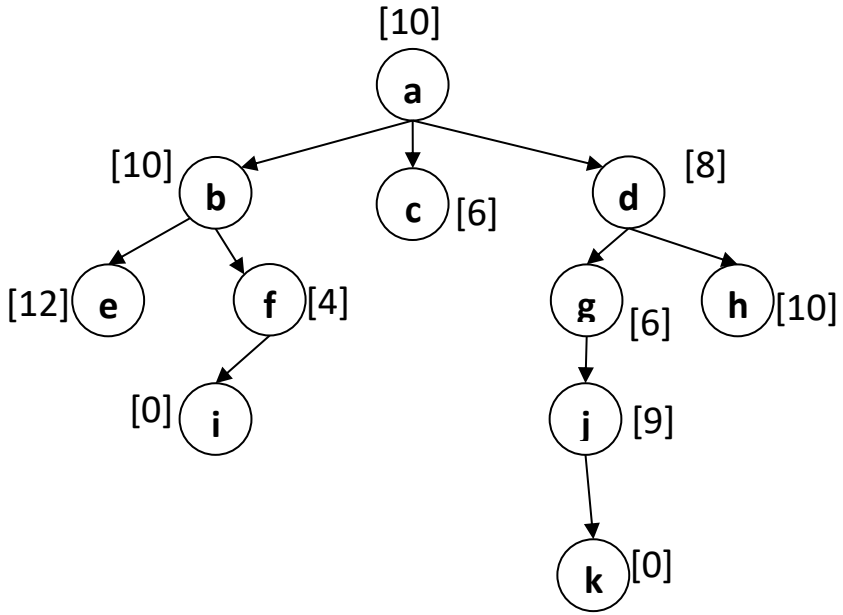
- Στο διπλανό σχήμα παρουσιάζεται ένα δένδρο αναζήτησης, όπου ο αριθμός μέσα σε αγκύλες δίπλα σε κάθε κόμβο αντιστοιχεί στην τιμή μιας Ευρετική ς συνάρτησης για αυτόν. Οι κόμβοι που έχουν τιμή 0, είναι οι τερματικοί κόμβοι της αναζήτησης.
- Γράψτε την σειρά με την οποία θα εξεταστούν οι κόμβοι του δέντρου (π.χ. a,b,c,…) μέχρι να βρεθεί τερματική κατάσταση από τους ακόλουθους αλγορίθμους:



- HC:……
- Best FS: ……
- A* : ……

Άσκηση 4.1 (Λύση HC & Best FS)

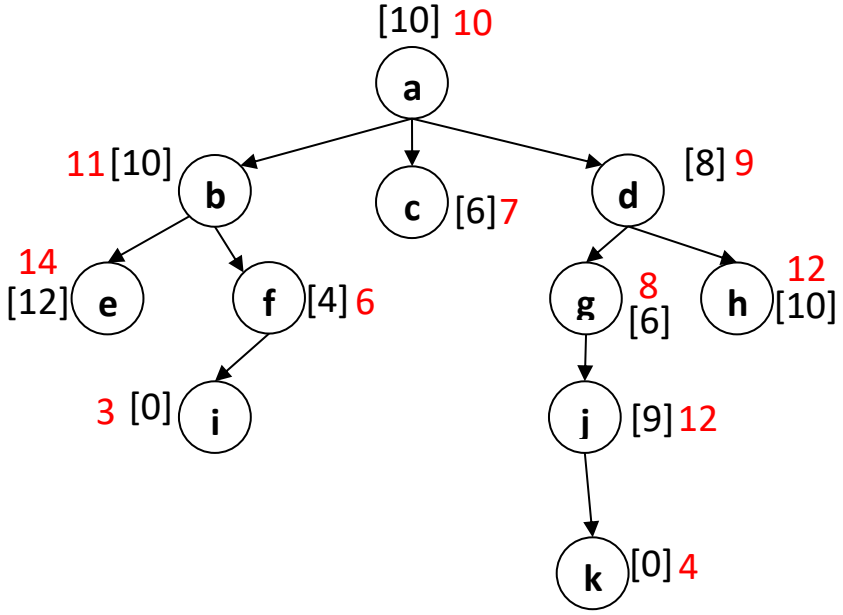
- Στο διπλανό σχήμα παρουσιάζεται ένα δένδρο αναζήτησης, όπου ο αριθμός μέσα σε αγκύλες δίπλα σε κάθε κόμβο αντιστοιχεί στην τιμή μιας Ευρετική ς συνάρτησης για αυτόν. Οι κόμβοι που έχουν τιμή 0, είναι οι τερματικοί κόμβοι της αναζήτησης.
- Γράψτε την σειρά με την οποία θα εξεταστούν οι κόμβοι του δέντρου (π.χ. a,b,c,…) μέχρι να βρεθεί τερματική κατάσταση από τους ακόλουθους αλγορίθμους:



- HC: a, c (αδιέξοδο)
- Best FS: a, c, d, g, j, k

Άσκηση 4.1 (Λύση A*)

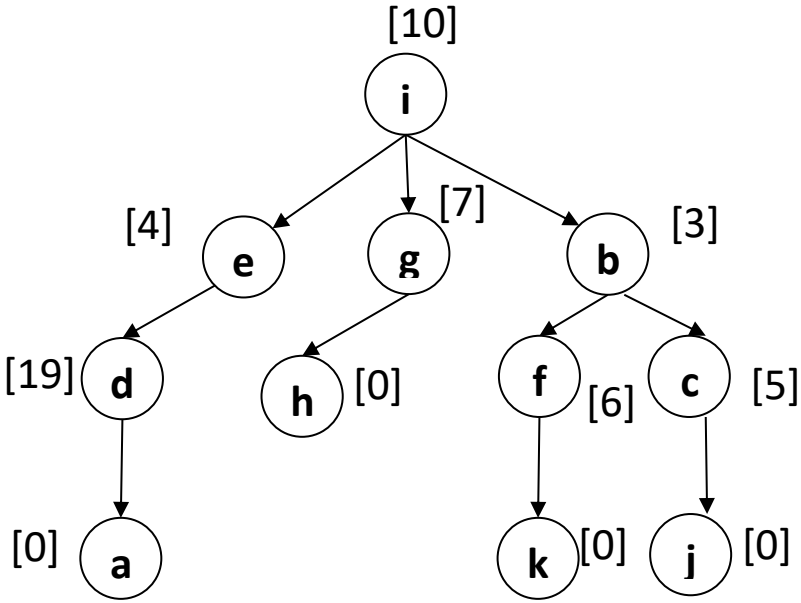
- Στο διπλανό σχήμα παρουσιάζεται ένα δένδρο αναζήτησης, όπου ο αριθμός μέσα σε αγκύλες δίπλα σε κάθε κόμβο αντιστοιχεί στην τιμή μιας Ευρετική ς συνάρτησης για αυτόν. Οι κόμβοι που έχουν τιμή 0, είναι οι τερματικοί κόμβοι της αναζήτησης.
- Γράψτε την σειρά με την οποία θα εξεταστούν οι κόμβοι του δέντρου (π.χ. a,b,c,…) μέχρι να βρεθεί τερματική κατάσταση από τους ακόλουθους αλγορίθμους:



- A* : a, c, d, g, b, f, i

Άσκηση 4.2

- Στο διπλανό σχήμα παρουσιάζεται ένα δένδρο αναζήτησης, όπου ο αριθμός μέσα σε αγκύλες δίπλα σε κάθε κόμβο αντιστοιχεί στην τιμή μιας Ευρετική ς συνάρτησης για αυτόν. Οι κόμβοι που έχουν τιμή 0, είναι οι τερματικοί κόμβοι της αναζήτησης.
- Γράψτε την σειρά με την οποία θα εξεταστούν οι κόμβοι του δέντρου (π.χ. a,b,c,…) μέχρι να βρεθεί τερματική κατάσταση από τους ακόλουθους αλγορίθμους:



- HC:……
- Best FS: ……
- A* : ……

Άσκηση 4.3

❖ Να προτείνετε έναν ευερετικό μηχανισμό στο παρακάτω πρόβλημα puzzle και να εφαρμόσετε τον αλγόριθμο Best First Search για την εύρεση των επόμενων 2 βημάτων.

IS:

2	5	1
3	6	4
7		8

FS:

1	2	3
4	5	6
7	8	

.....

.....

.....

.....

.....

Άσκηση 4.3 (Λύση)

- Να προτείνετε έναν ευερετικό μηχανισμό στο παρακάτω πρόβλημα puzzle και να εφαρμόσετε τον αλγόριθμο Best First Search για την εύρεση των επόμενων 2 βημάτων.

IS:

2	5	1
3	6	4
7		8

FS:

1	2	3
4	5	6
7	8	

- Ευριστική: Πλήθος πλακιδίων εκτός θέσης
  - Από IS υπάρχουν 3 κινήσεις:
    - ✓ S(1) Το κενό να μετακινηθεί αριστερά h(S1) = 8
    - ✓ S(2) Το κενό να μετακινηθεί δεξιά h(S2) = 5
    - ✓ S(3) Το κενό να μετακινηθεί πάνω h(S3) = 7
  - Επιλέγω S(2)
  - Από S(2 υπάρχουν 2 κινήσεις:
    - ✓ S(4) Το κενό να μετακινηθεί αριστερά (IS - απορρίπτεται)
    - ✓ S(5) Το κενό να μετακινηθεί πάνω h(S5) = 6
  - Επιλέγω S(5)

S2:

2	5	1
3	6	4
7	8	

FS:

1	2	3
4	5	6
7	8	

S5:

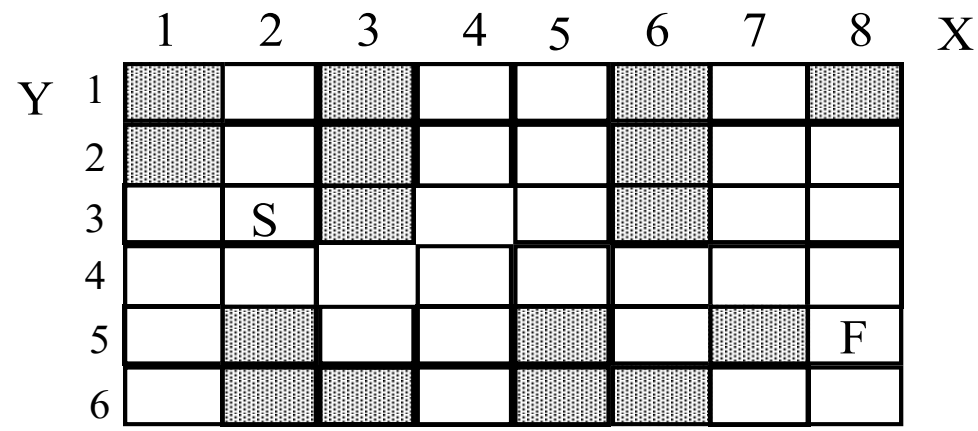
2	5	1
3	6	
7	8	4

FS:

1	2	3
4	5	6
7	8	

Άσκηση 4.4

- Να εφαρμόσετε έναν ευερετικό μηχανισμό στο ακόλουθο πρόβλημα του λαβυρίνθου και να προτείνετε την επόμενη κίνηση από την αρχική θέση S (σημείωση: διαγώνια κίνηση δεν επιτρέπεται). (0,5 Μ)



.....

.....

.....

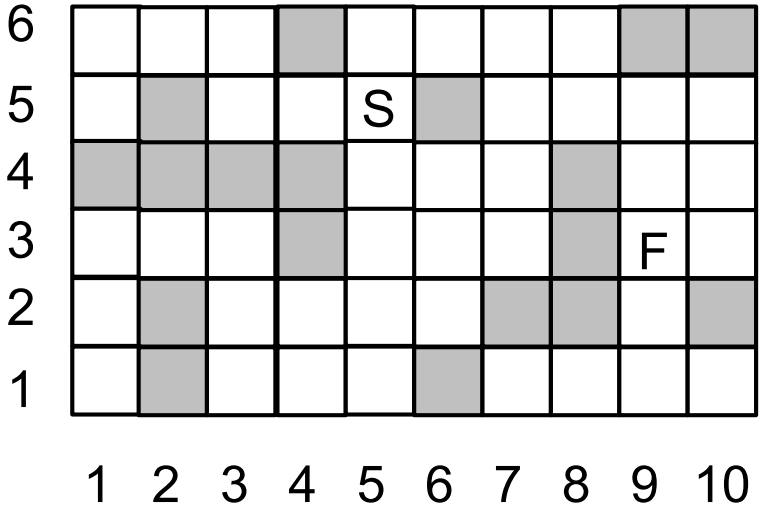
.....

.....

.....

Άσκηση 4.5

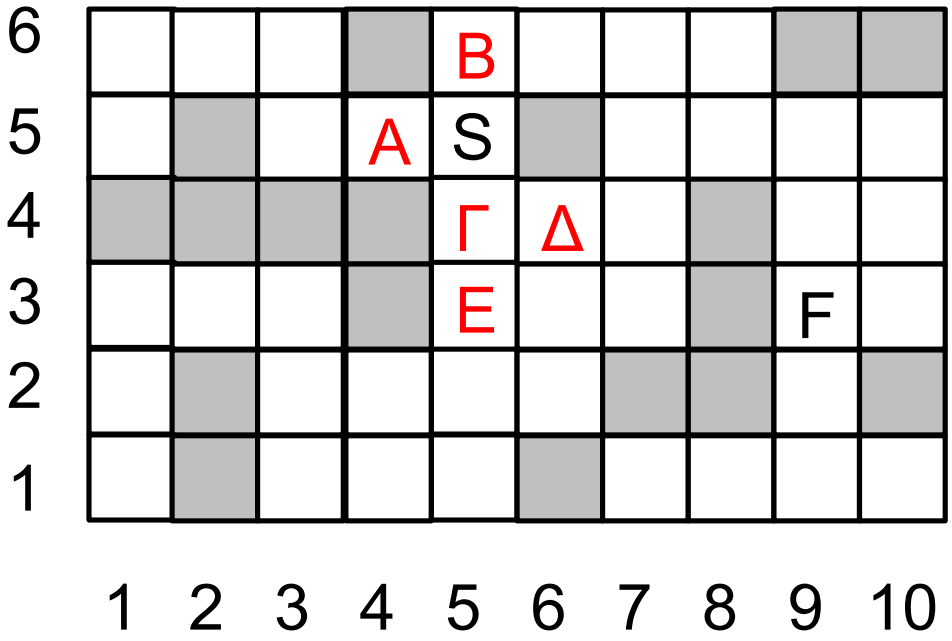
- Να εφαρμοστεί ο BestFS στο λαβύ-ρινθο, στον οποίο ζητείται μία διαδρομή από το S στο F, χρησιμοποιώντας ως ευρετική συνάρτηση την απόσταση Manhattan.
- Να βρεθούν 2 βήματα και να συμπληρωθεί ο πίνακας, γράφοντας την τιμή της συνάρτησης ως εκθέτη.



Μέτωπο Αναζήτησης	Κλειστό Σύνολο	Κατάσταση	Παιδιά

Άσκηση 4.5 (Λύση)

- Να εφαρμοστεί ο BestFS στο λαβύ-ρινθο, στον οποίο ζητείται μία διαδρο-μή από το S στο F, χρησιμοποιώντας ως ευρετική συνάρτηση την απόσταση Manhattan.
- Να βρεθούν 2 βήματα και να συμπληρωθεί ο πίνακας, γράφοντας την τιμή της συνάρτησης ως εκθέτη.

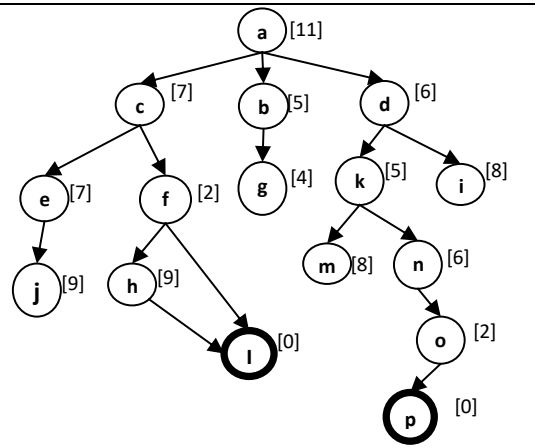


Μέτωπο Αναζήτησης	Κλειστό Σύνολο	Κατάσταση	Παιδιά
$\{S^6\}$	$\varnothing$	S^6	A^7, B^7, Γ^5
$\{\Gamma^5, A^7, B^7\}$	$\{S\}$	Γ^5	Δ^4, E^4, S^6

- Στο διπλανό σχήμα παρουσιάζεται ένα δένδρο αναζήτησης, όπου ο αριθμός μέσα σε αγκύλες δίπλα σε κάθε κόμβο αντιστοιχεί στην τιμή μιας ευριστικής συνάρτησης.
- Οι κόμβοι που έχουν τιμή 0, είναι οι τερματικοί κόμβοι της αναζήτησης.
- Γράψτε:

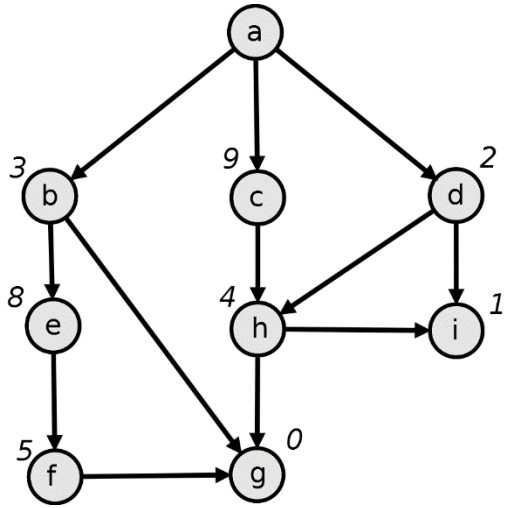
  α) την σειρά με την οποία θα εξεταστούν οι κόμβοι του δέντρου (π.χ. a,b,c,…) και

  β) τη λύση που θα επιστρέψουν (π.χ. <a,b,c,…>) μέχρι να βρεθεί τερματική κατάσταση από τους ακόλουθους αλγορίθμους:



Αλγόριθμος	Κόμβοι που εξετάστηκαν	Λύση
DFS	a,c,e,j,f,h,l	<a,c,f,h,l>
BFS	a,c,b,d,e,f,g,k,i,j,h,l	<a,c,f,l>
HC	a,b,g	∅
BestFS	a,b,g,d,k,n,o,p	<a,d,k,n,o,p>
A*	a,b,g,d,k,c,f,l	<a,c,f,l>

- Δίνεται ο παρακάτω χώρος αναζήτησης όπου a η αρχική κατάσταση και g η τελική κατάσταση. Οι αριθμοί δίπλα σε κάθε κατάσταση δείχνουν την ευρετική τιμή της (όσο χαμηλότερη τιμή τόσο καλύτερη η κατάσταση).



- Γράψτε τη σειρά με την οποία θα επισκεφθεί τις καταστάσεις ο αλγόριθμος στην προσπάθεια να βρει μία λύση στο πρόβλημα:
  - α) Αναζήτηση με Αναρρίχηση Λόφων (HC)
  - β) Αναζήτηση Πρώτα στο Καλύτερο (BestFS)
  - γ) Αναζήτηση A*

- Έστω το διπλανό παιχνίδι τρίλιζας στο οποίο είναι η σειρά του Η/Υ να παίξει (να βάλει δηλαδή ένα Ο σε ένα από τα λευκά κελιά).
- Να προτείνετε μία ευρετική συνάρτηση και να την εφαρμόσετε στον ΗC ώστε να επιλεχθεί η επόμενη κίνηση

X	O	X
	X	
O		