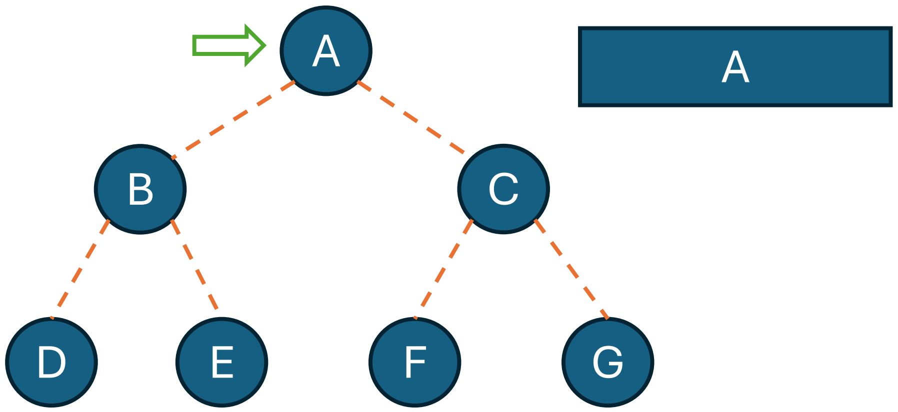
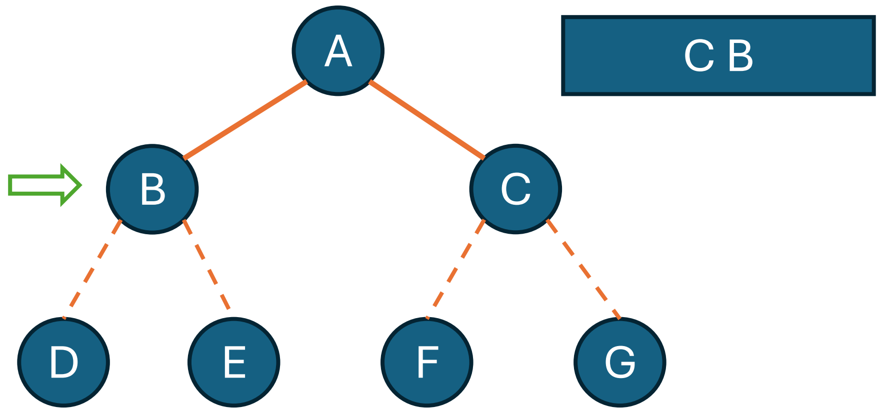
Τεχνητή νοημοσύνη

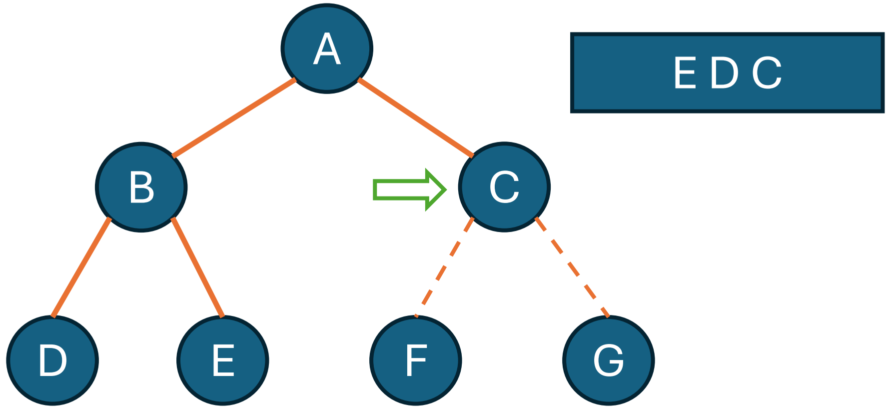
Φροντιστήριο 2

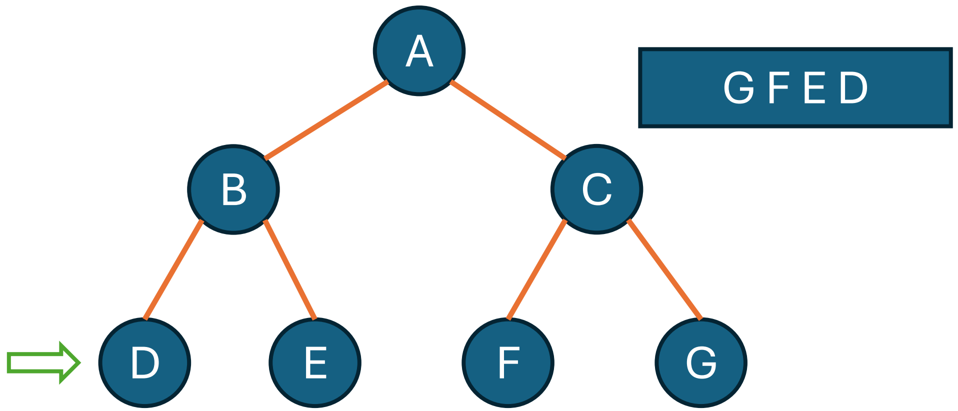
Ασκήσεις μελέτης της 3ης και 4ης διάλεξης

Breadth-First Search (BFS)

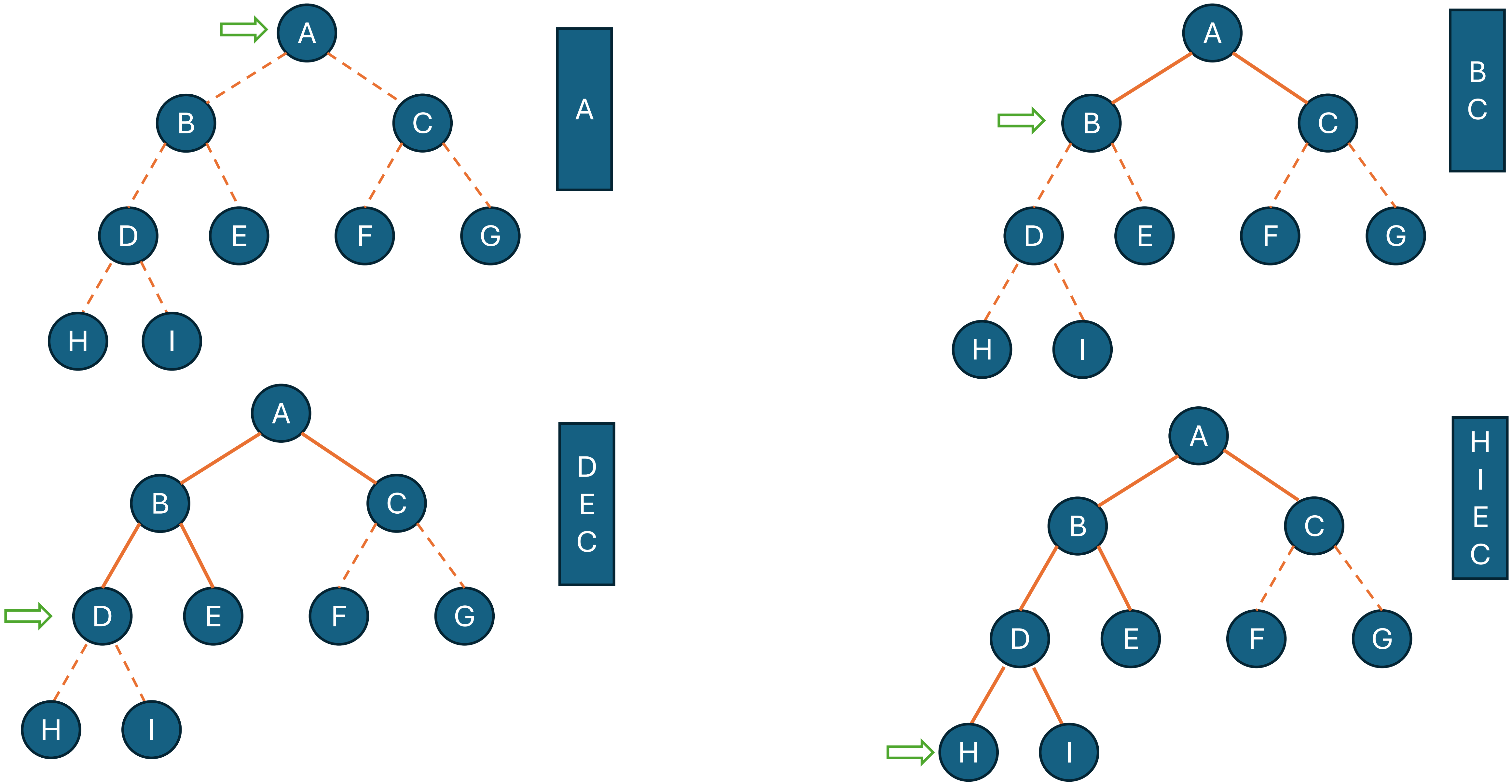




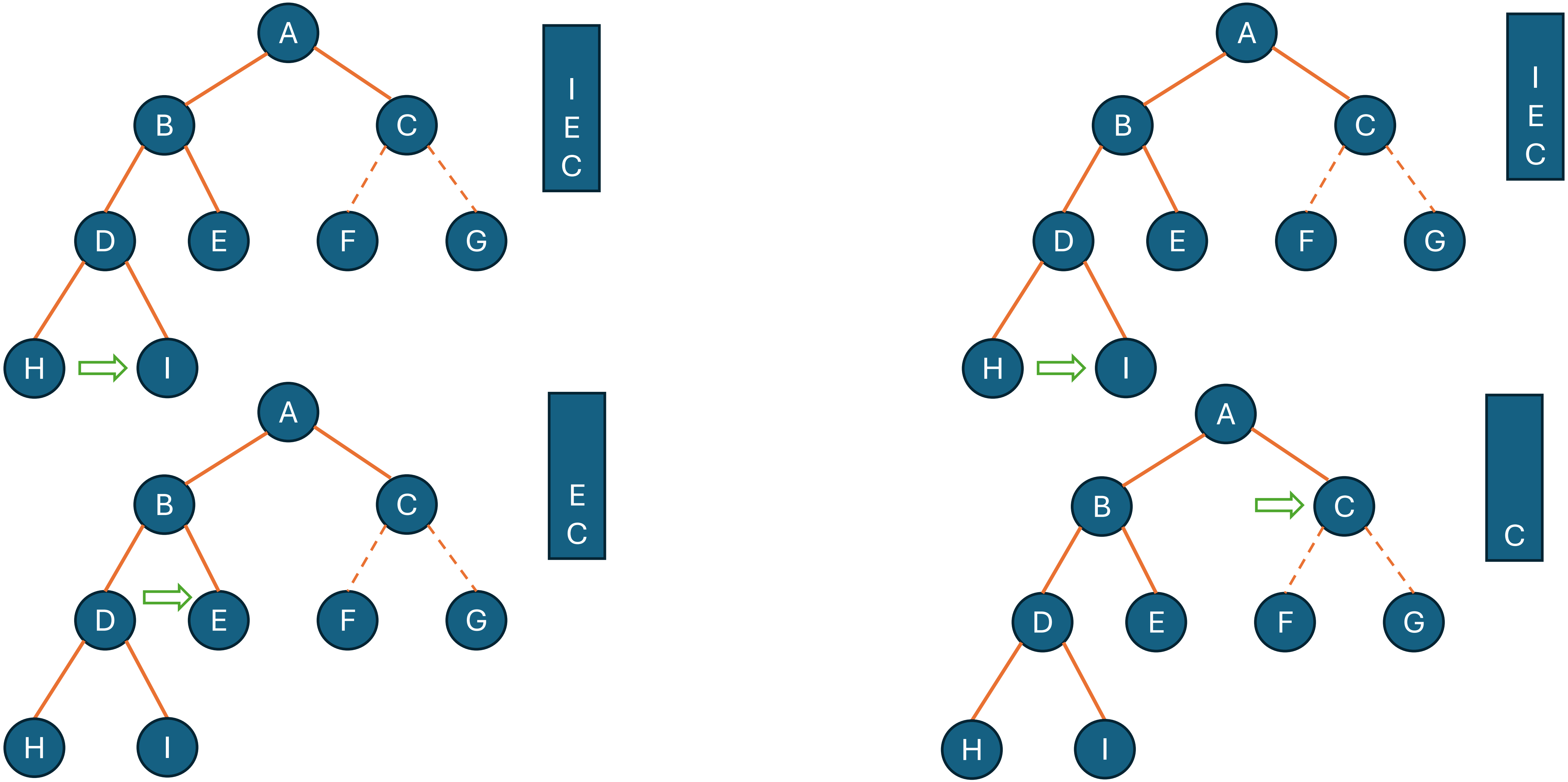




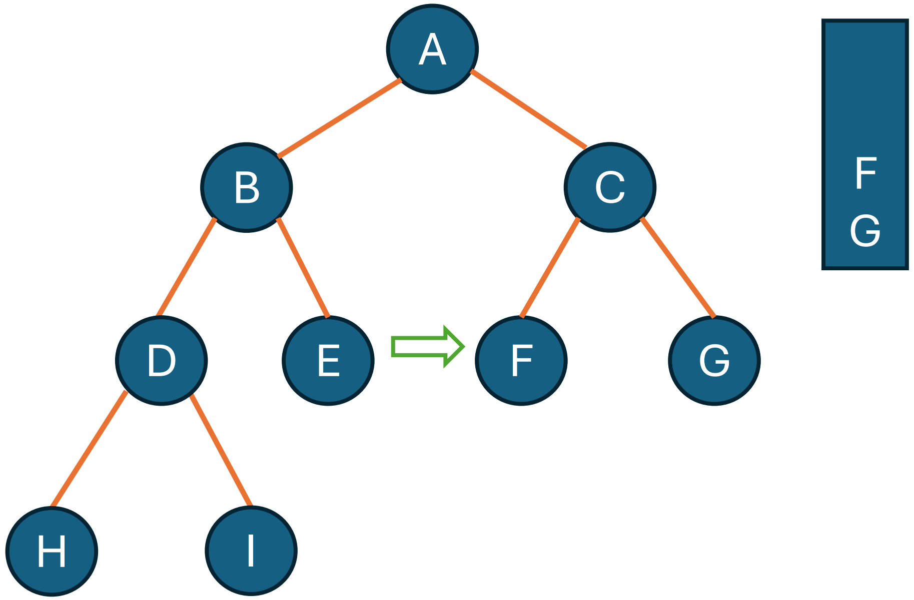
Depth-First Search (DFS)

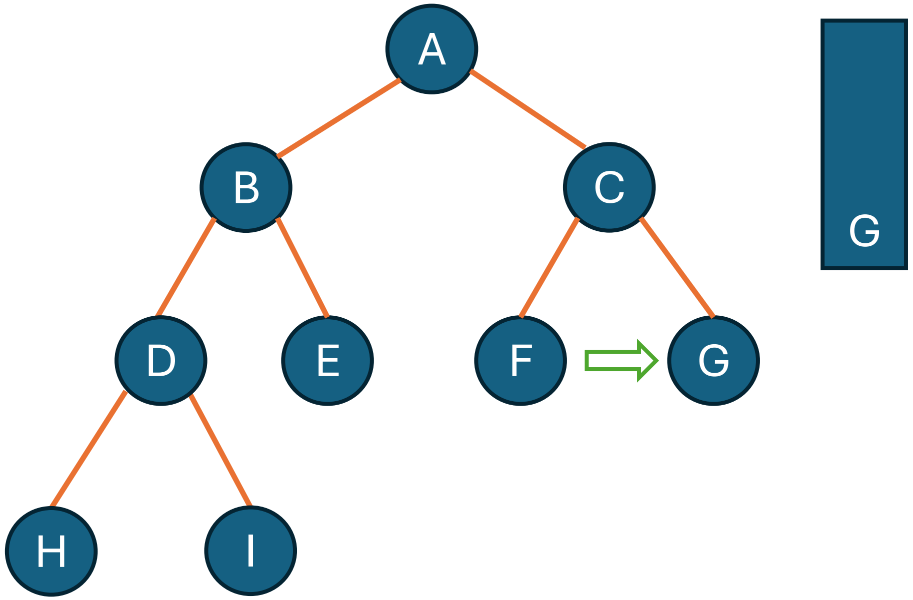


Depth-First Search (DFS)



Depth-First Search (DFS)





Άσκηση 3.1(α)

Ο DFS έχει:

α) χειρότερη

β) καλύτερη

γ) την ίδια

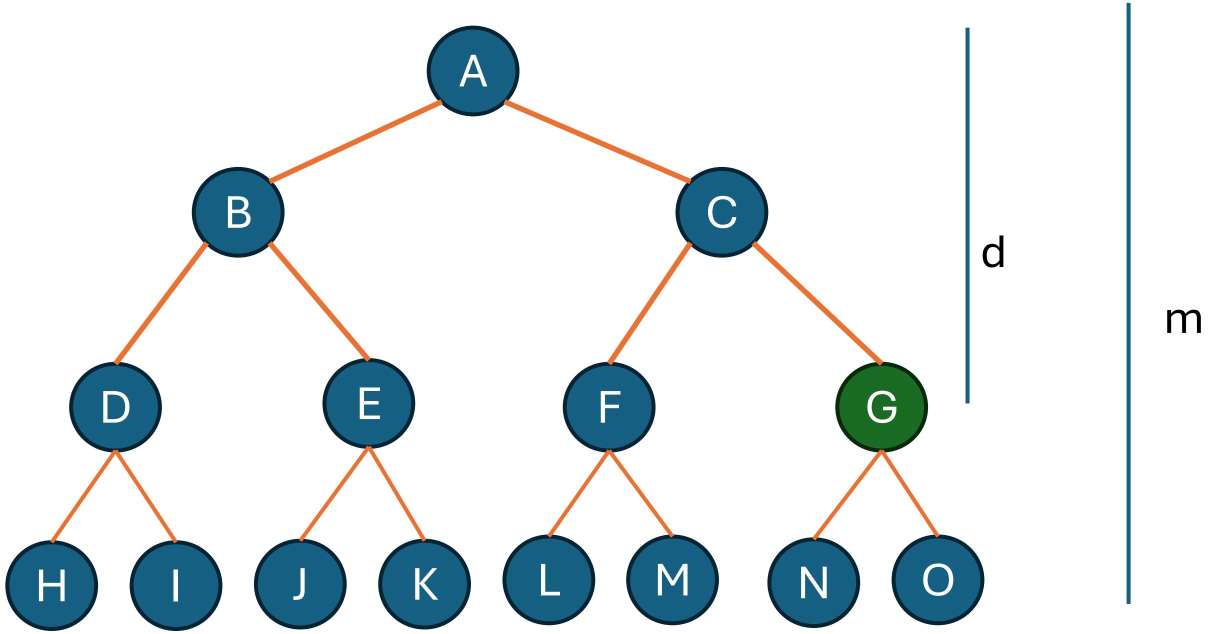
πολυπλοκότητα χρόνου σε σχέση με τον BFS

Breadth-First Search (BFS)

Μέγιστος παράγοντας διακλάδωσης (b) = Μέγιστος δυνατός αριθμών παιδιών που προκύπτουν από την επέκταση ενός κόμβου

Βάθος ρηχότερης λύσης (d)

Μέγιστο δυνατό βάθος (m)



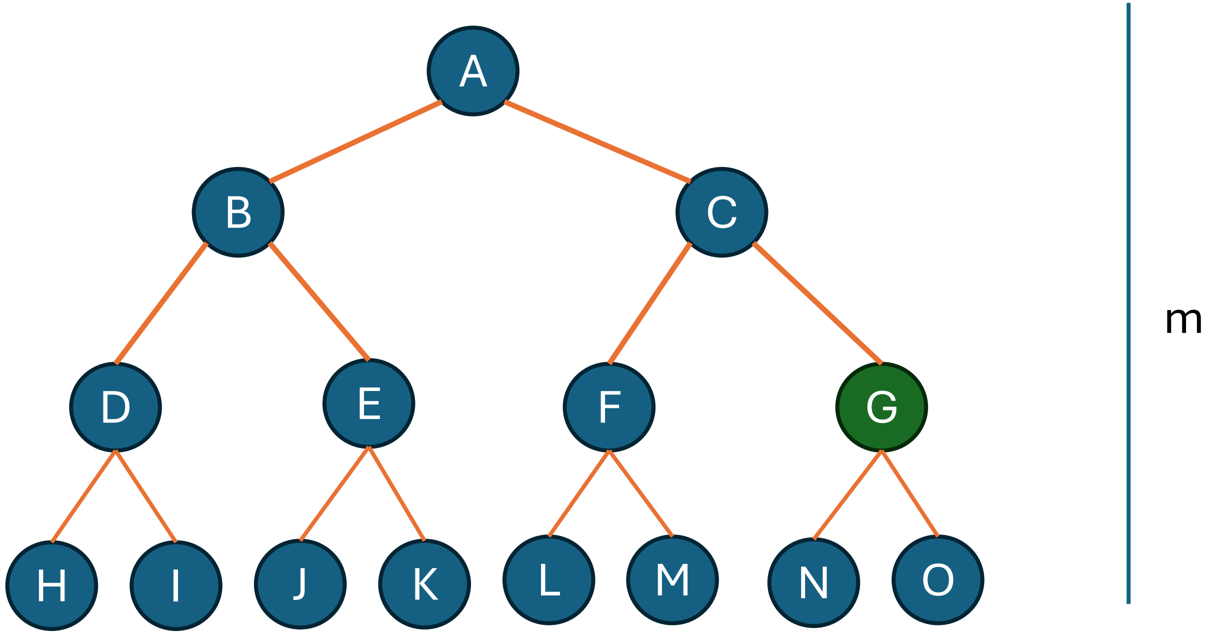
Πολυπλοκότητα χρόνου: $O(b^{d+1})$

Αν b πεπερασμένο και το κόστος λύσης είναι αύξουσα συνάρτηση του βάθους BFS βέλτιστος

Depth-First Search (DFS)

Μέγιστος παράγοντας διακλάδωσης (b) = Μέγιστος δυνατός αριθμών παιδιών που προκύπτουν από την επέκταση ενός κόμβου

Μέγιστο δυνατό βάθος (m)



Πολυπλοκότητα χρόνου: $O(b^m)$

Μπορεί να μη βρει μια εναλλακτική τελική κατάσταση σε μικρότερο βάθος. DFS μη πλήρης και μη βέλτιστος.

Άσκηση 3.1(α)

Ο DFS έχει:

<u>α) χειρότερη</u>

β) καλύτερη

γ) την ίδια

πολυπλοκότητα χρόνου σε σχέση με τον BFS

Άσκηση 3.1(β)

Ο DFS έχει:

α) χειρότερη

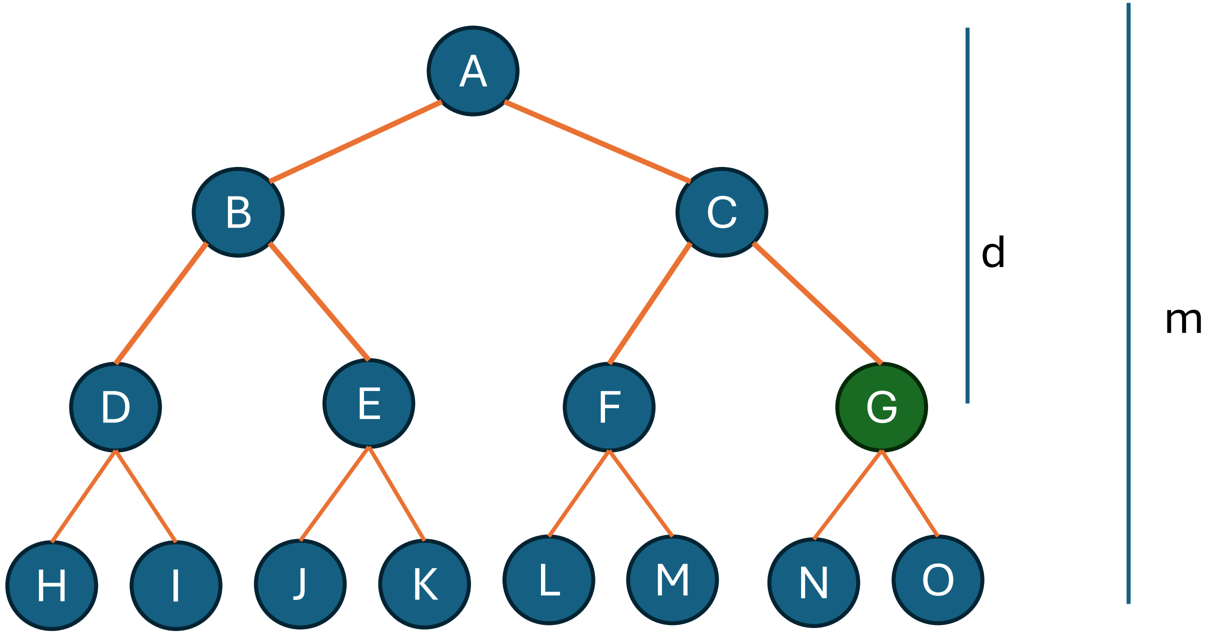
β) καλύτερη

γ) την ίδια

πολυπλοκότητα χώρου σε σχέση με τον BFS

Breadth-First Search (BFS)

Μέγιστος παράγοντας διακλάδωσης (b) = Μέγιστος δυνατός αριθμών παιδιών που προκύπτουν από την επέκταση ενός κόμβου



Βάθος ρηχότερης λύσης (d)

Μέγιστο δυνατό βάθος (m)

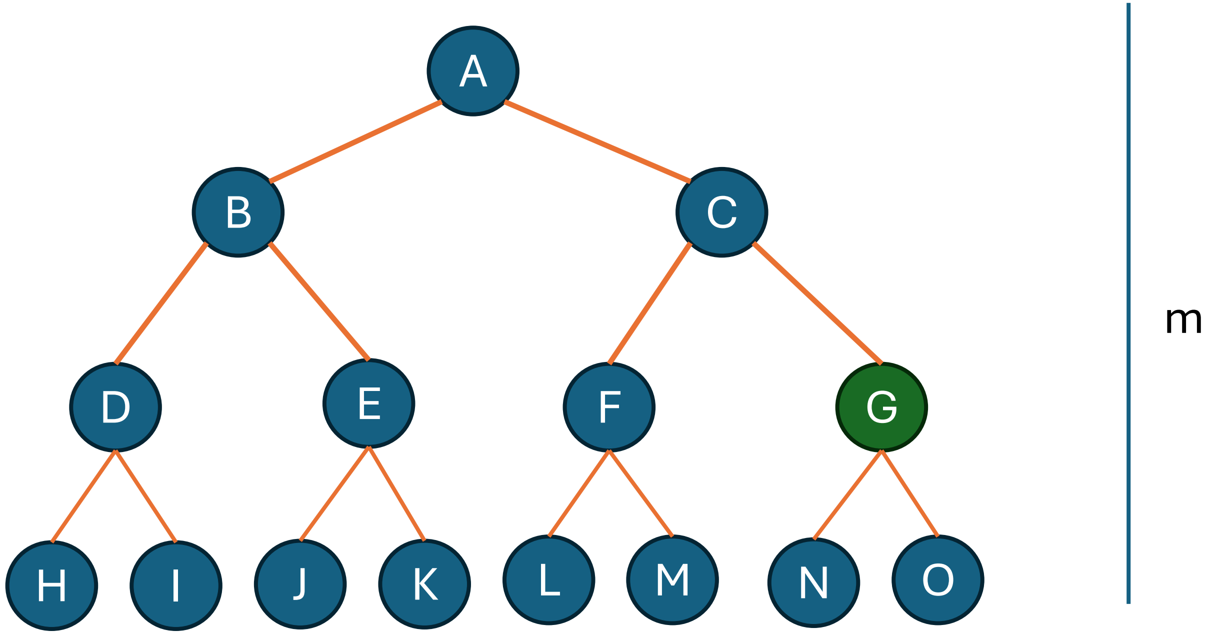
Πολυπλοκότητα χώρου : $O(b^{d+1})$

Αν b πεπερασμένο και το κόστος λύσης είναι αύξουσα συνάρτηση του βάθους BFS βέλτιστος

Depth-First Search (DFS)

Μέγιστος παράγοντας διακλάδωσης (b) = Μέγιστος δυνατός αριθμών παιδιών που προκύπτουν από την επέκταση ενός κόμβου

Μέγιστο δυνατό βάθος (m)



Μπορεί να μη βρει μια εναλλακτική τελική κατάσταση σε μικρότερο βάθος. DFS μη πλήρης και μη βέλτιστος.

Πολυπλοκότητα χώρου : $O(bm)$ -- αποθηκεύουμε τα παιδιά όλων των προγόνων του φύλλου και τη ρίζα, δηλαδή bm + 1 κόμβους.

Άσκηση 3.1(β)

Ο DFS έχει:

α) χειρότερη
β) καλύτερη
γ) την ίδια

πολυπλοκότητα χώρου σε σχέση με τον BFS

(βέβαια για άπειρο m ενδέχεται να μην τερματίσει)

Άσκηση 3.1(γ)

1. Γνωρίζουμε ότι **υπάρχει λύση**
2. Το **b είναι πεπερασμένο**
3. Το **σύνολο καταστάσεων είναι πεπερασμένο**
4. Το **κόστος λύσης είναι αύξουσα συνάρτηση του βάθους** (και μόνο)

Ο DFS με **κλειστό σύνολο**:

α) βρίσκει πάντα λύση και μάλιστα βέλτιστη

β) δεν βρίσκει πάντα λύση

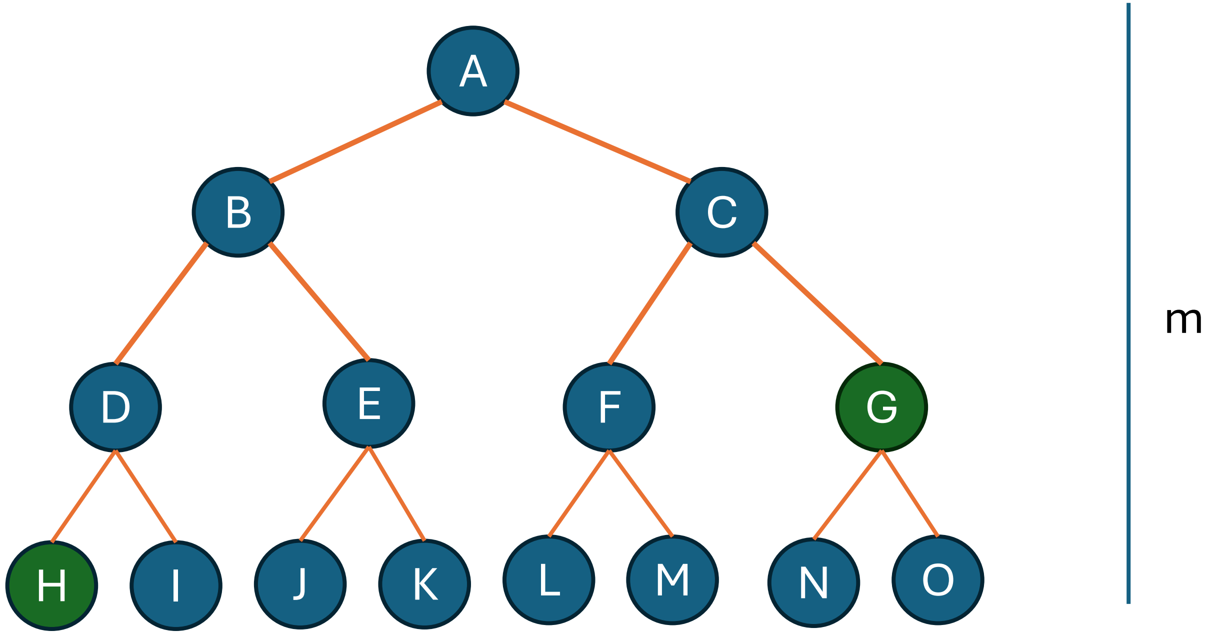
γ) βρίσκει πάντα λύση, αλλά όχι σίγουρα βέλτιστη

Άσκηση 3.1(γ)

Λύση: Αφού το σύνολο καταστάσεων είναι πεπερασμένο, τα μόνα άπειρα κλαδιά που είναι δυνατόν να προκύψουν αντιστοιχούν σε κύκλους του γράφου καταστάσεων. Αφού, όμως, χρησιμοποιείται κλειστό σύνολο, ο DFS δεν θα παγιδευτεί ποτέ σε τέτοια άπειρα κλαδιά. Επομένως, αφού και ο μέγιστος παράγοντας διακλάδωσης είναι πεπερασμένος, το δέντρο αναζήτησης που θα ψάξει ο DFS είναι πεπερασμένο. Επειδή υπάρχει λύση, ο DFS θα τερματίσει συναντώντας τελική κατάσταση. Η λύση, όμως, που θα αναφέρει ενδέχεται να μην είναι βέλτιστη, γιατί ενδέχεται να υπάρχει και άλλη τελική κατάσταση σε μικρότερο βάθος

Άσκηση 3.1(γ)

Αν πχ τα H και G είναι τελικές καταστάσεις, ο DFS θα επιστέψει το μονοπάτι για τον H παρόλο που το G είναι στη βέλτιστη λύση



Άσκηση 3.1(γ)

1. Γνωρίζουμε ότι **υπάρχει λύση**
2. Το **b είναι πεπερασμένο**
3. Το **σύνολο καταστάσεων είναι πεπερασμένο**
4. Το **κόστος λύσης είναι αύξουσα συνάρτηση του βάθους** (και μόνο)

Ο DFS με **κλειστό σύνολο**:

α) βρίσκει πάντα λύση και μάλιστα βέλτιστη

β) δεν βρίσκει πάντα λύση

<u>γ) βρίσκει πάντα λύση, αλλά όχι σίγουρα βέλτιστη</u>

Άσκηση 3.1(δ)

1. **ΔΕΝ υπάρχει λύση**
2. Το **b είναι πεπερασμένο**
3. Το **σύνολο καταστάσεων είναι πεπερασμένο**

Ο DFS με **κλειστό σύνολο**:

α) τερματίζει πάντα

β)δεν τερματίζει ποτέ

γ) άλλοτε τερματίζει και άλλοτε δεν τερματίζει

Άσκηση 3.1(δ)

Λύση: Όπως και στο προηγούμενο ερώτημα, το δέντρο αναζήτησης που θα ψάξει ο DFS είναι πεπερασμένο. Αφού τώρα σίγουρα δεν υπάρχει λύση, άρα δεν υπάρχει και τελική κατάσταση, ο DFS απλά θα ψάξει ολόκληρο το (πεπερασμένο) δέντρο αναζήτησης και θα τερματίσει.

Άσκηση 3.1(δ)

1. **ΔΕΝ υπάρχει λύση**
2. Το **b είναι πεπερασμένο**
3. Το **σύνολο καταστάσεων είναι πεπερασμένο**

Ο DFS με **κλειστό σύνολο**:

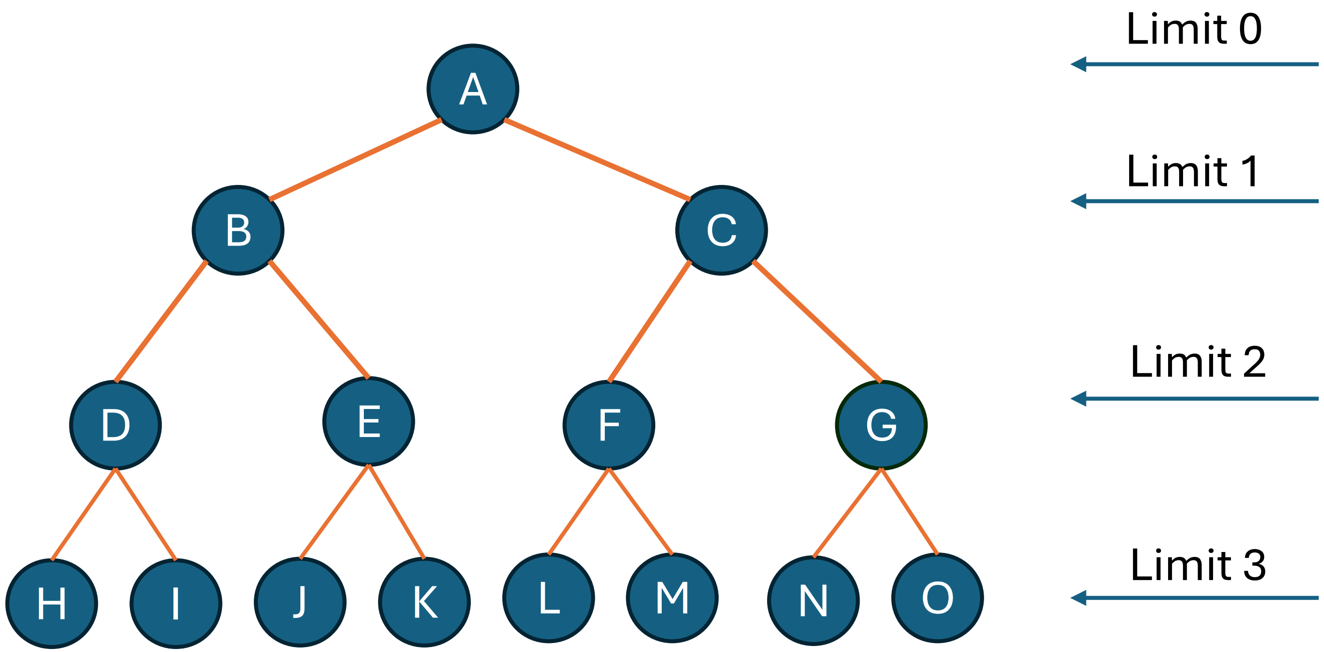
<u>α) τερματίζει πάντα</u>

β)δεν τερματίζει ποτέ

γ) άλλοτε τερματίζει και άλλοτε δεν τερματίζει

Iterative Deepening Search (IDS)

Ξεκινάμε από βάθος 0, κάνουμε DFS αυξάνοντας το βάθος μέχρι να βρούμε λύση



Άσκηση 3.1(ε)

Ο IDS χωρίς κλειστό σύνολο έχει:

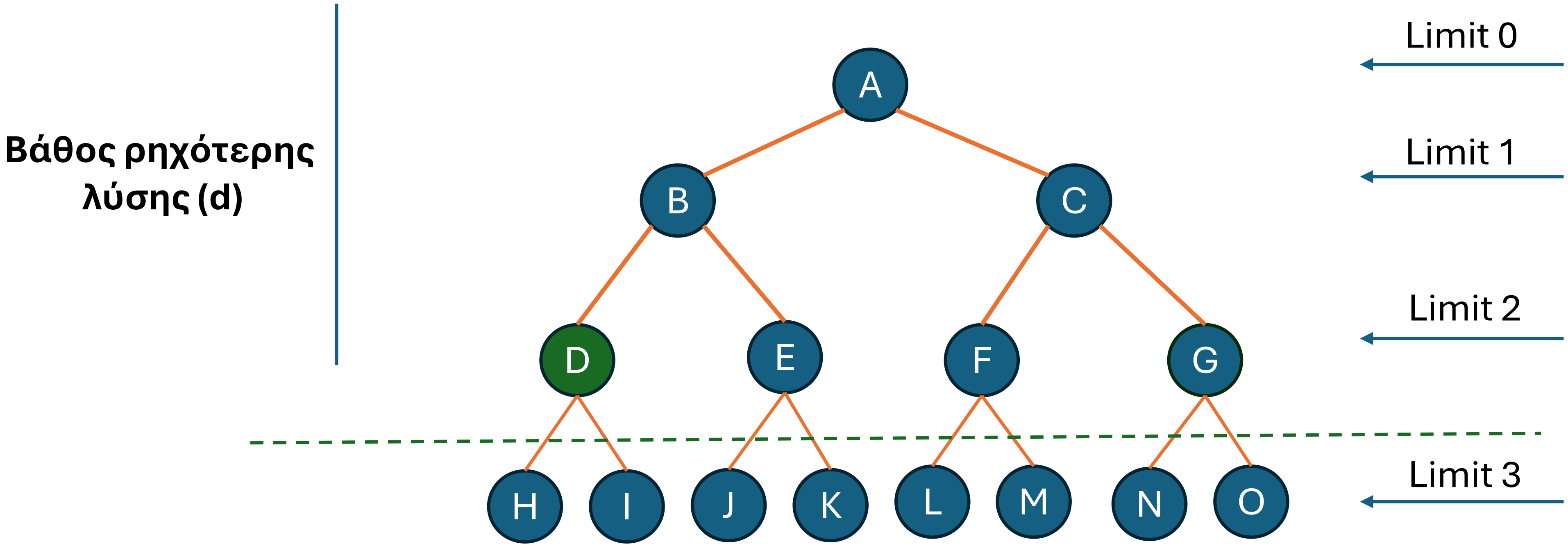
α) καλύτερη

β) χειρότερη

γ) την ίδια

πολυπλοκότητα χώρου συγκρινόμενος με τον **BFS χωρίς κλειστό σύνολο.**

Iterative Deepening Search (IDS)



Πολυπλοκότητα χώρου: $O(bm)$ – δεν υπερβαίνουμε το βάθος της ρηχότερης λύσης

Άσκηση 3.1(ε)

Ο IDS χωρίς κλειστό σύνολο έχει:

α) καλύτερη

β) χειρότερη

γ) την ίδια

πολυπλοκότητα χώρου συγκρινόμενος με τον **BFS χωρίς κλειστό σύνολο.**

Άσκηση 3.1(στ)

Ο IDS χωρίς κλειστό σύνολο έχει:

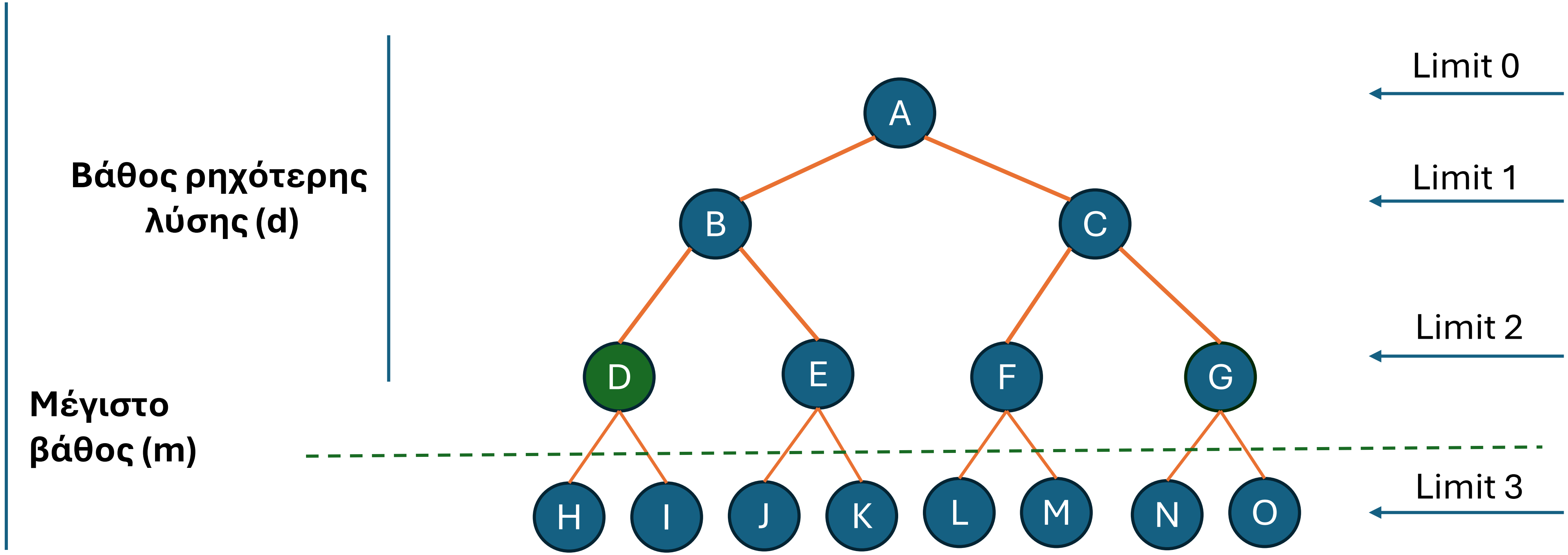
α) καλύτερη

β) χειρότερη

γ) την ίδια

πολυπλοκότητα χώρου συγκρινόμενος με τον **DFS χωρίς κλειστό σύνολο.**

Iterative Deepening Search (IDS)



Πολυπλοκότητα χώρου: $O(bm)$ – δεν υπερβαίνουμε το βάθος της ρηχότερης λύσης

O DFS ψάχνει μέχρι το μέγιστο βάθος m

Άσκηση 3.1(στ)

Ο IDS χωρίς κλειστό σύνολο έχει:

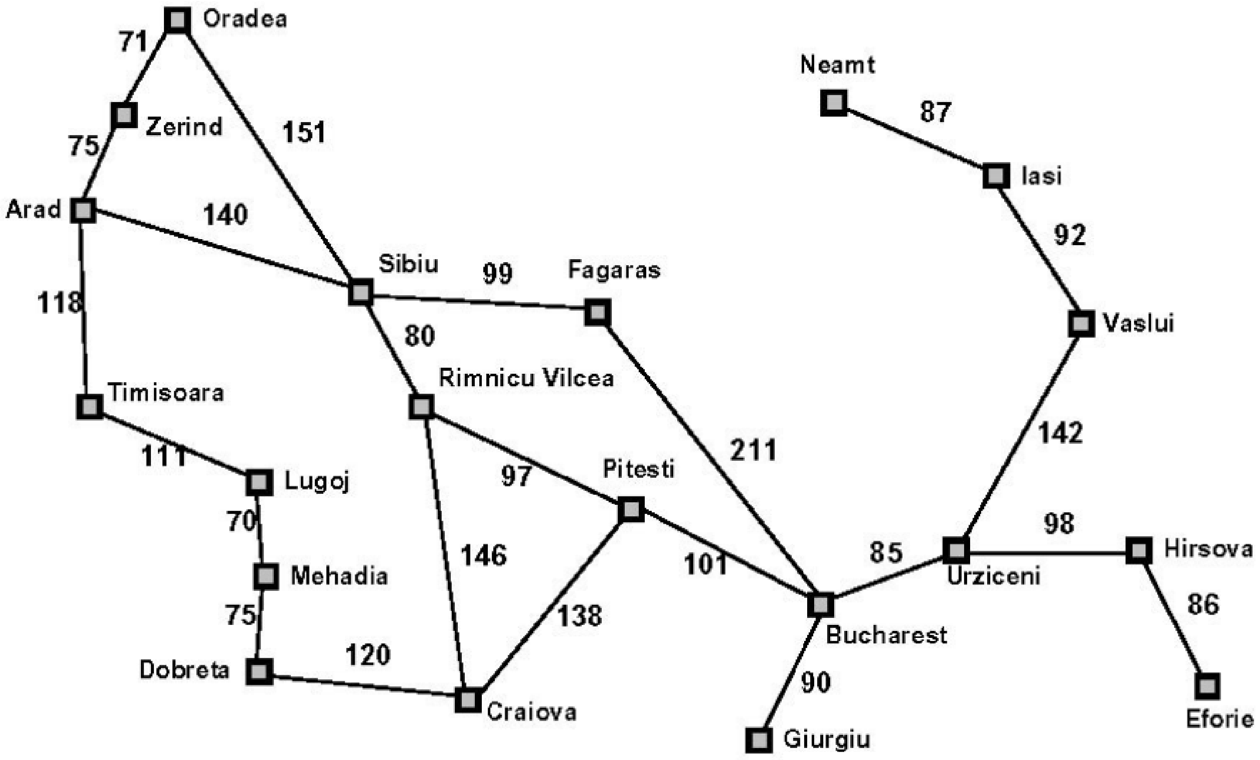
α) καλύτερη (d ≤m)

β) χειρότερη

γ) την ίδια

πολυπλοκότητα χώρου συγκρινόμενος με τον **DFS χωρίς κλειστό σύνολο.**

Άσκηση 4.2(α)

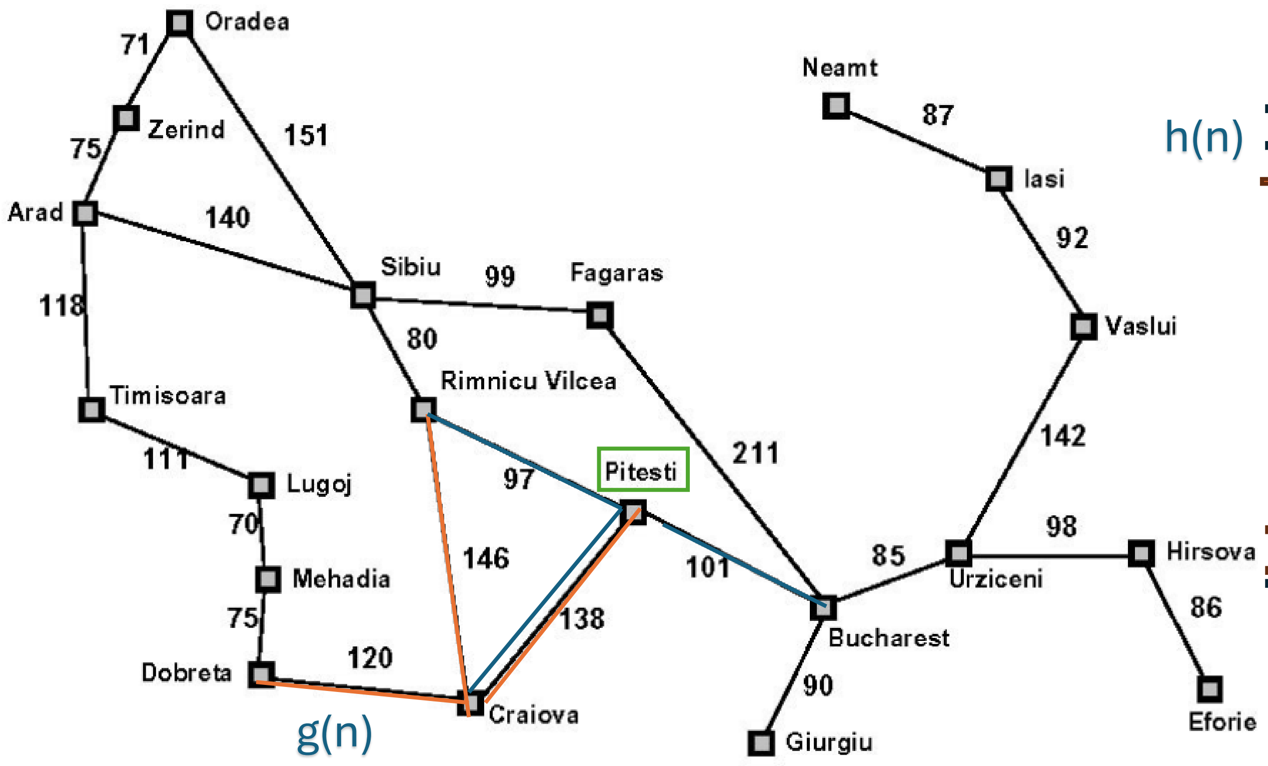


Straight−line distance
to Bucharest

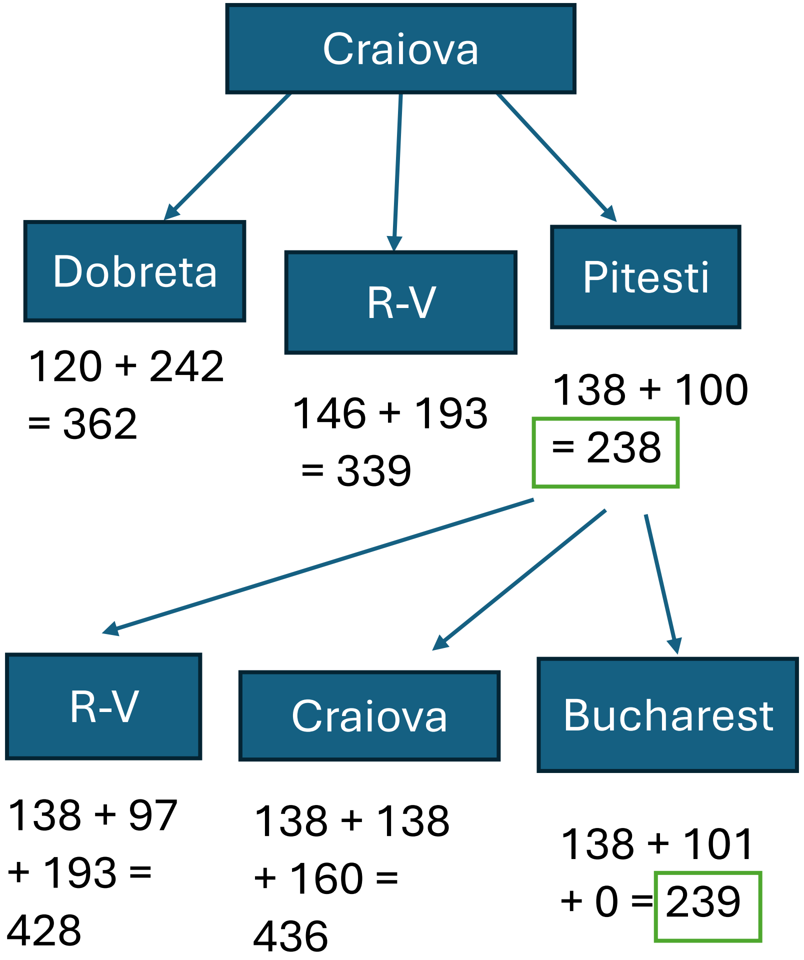
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Σχεδιάστε το δέντρο αναζήτησης που κατασκευάζει ο A* μέχρι να ανακαλύψει το πρώτο μονοπάτι από την Craiova στο Βουκουρέστι.

Άσκηση 4.2(α)



Straight−line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374



f(n) = h(n) + g(n)

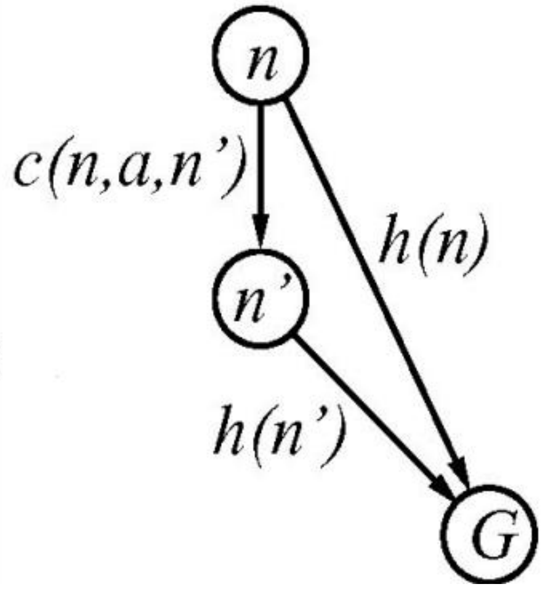
Αποδεκτή και συνεπής ευρετική

Αποδεκτή : **h(n) ≤ C(n)***

(πραγματικό κόστος βέλτιστου μονοπατιού από τον n σε τελική κατάσταση)

Τελεστής που οδηγεί από το n στο n’

Συνεπής: **h(n) ≤ c(n, a, n') + h(n’)**



Κάθε συνεπής είναι και αποδεκτή

Άσκηση 4.2(β)

Χρησιμοποιούμε την ευθεία απόσταση μέχρι την πόλη στόχο

Είναι αποδεκτή η ευρετική που χρησιμοποιούμε; Ναι ή όχι και γιατί;

Άσκηση 4.2(β)

Αποδεκτή : **h(n) ≤ C(n)***

(πραγματικό κόστος βέλτιστου μονοπατιού από τον n σε τελική κατάσταση)

Λύση: Ναι είναι αποδεκτή γιατί η ευθεία απόσταση μέχρι την πόλη στόχο είναι υποεκτίμηση της πραγματικής (οδικής) απόστασης.

Άσκηση 4.2(γ)

Μπορούμε να είμαστε σίγουροι ότι ο αλγόριθμος θα ανακαλύπτει πάντα το **συντομότερο μονοπάτι** προς το Βουκουρέστι, **από όποια πόλη του χάρτη και αν ξεκινήσουμε**; Ναι ή όχι και γιατί;

Άσκηση 4.2(γ)

Μπορούμε να είμαστε σίγουροι ότι ο αλγόριθμος θα ανακαλύπτει πάντα το **συντομότερο μονοπάτι** προς το Βουκουρέστι, **από όποια πόλη του χάρτη και αν ξεκινήσουμε**; Ναι ή όχι και γιατί;

Είναι ο Α* πλήρης;

Άσκηση 4.2(γ)

Μπορούμε να είμαστε σίγουροι ότι ο αλγόριθμος θα ανακαλύπτει πάντα το **συντομότερο μονοπάτι** προς το Βουκουρέστι, **από όποια πόλη του χάρτη και αν ξεκινήσουμε**; Ναι ή όχι και γιατί;

Είναι ο A* πλήρης;

Λύση: Τα κόστη των μεταβάσεων είναι πάντα θετικά, το b είναι πεπερασμένο και ξέρουμε ότι τότε ο A* είναι πλήρης, δηλαδή βρίσκει πάντα λύση αν υπάρχει.

Άσκηση 4.2(γ)

Μπορούμε να είμαστε σίγουροι ότι ο αλγόριθμος θα ανακαλύπτει πάντα το **συντομότερο μονοπάτι** προς το Βουκουρέστι, **από όποια πόλη του χάρτη και αν ξεκινήσουμε**; Ναι ή όχι και γιατί;

Είναι ο A* πλήρης;

Λύση: Τα κόστη των μεταβάσεων είναι πάντα θετικά, το b είναι πεπερασμένο και ξέρουμε ότι τότε ο A* είναι πλήρης, δηλαδή βρίσκει πάντα λύση αν υπάρχει.

Υπάρχει πάντα λύση;

Άσκηση 4.2(γ)

Μπορούμε να είμαστε σίγουροι ότι ο αλγόριθμος θα ανακαλύπτει πάντα το **συντομότερο μονοπάτι** προς το Βουκουρέστι, **από όποια πόλη του χάρτη και αν ξεκινήσουμε**; Ναι ή όχι και γιατί;

Είναι ο A* πλήρης;

Λύση: Τα κόστη των μεταβάσεων είναι πάντα θετικά, το b είναι πεπερασμένο και ξέρουμε ότι τότε ο A* είναι πλήρης, δηλαδή βρίσκει πάντα λύση αν υπάρχει.

Υπάρχει πάντα λύση;

Λύση: Εδώ υπάρχει πάντα λύση (υπάρχει πάντα ένα μονοπάτι μέχρι το Βουκουρέστι), επομένως βρίσκει πάντα λύση.

Άσκηση 4.2(γ)

Μπορούμε να είμαστε σίγουροι ότι ο αλγόριθμος θα ανακαλύπτει πάντα το **συντομότερο μονοπάτι** προς το Βουκουρέστι, **από όποια πόλη του χάρτη και αν ξεκινήσουμε**; Ναι ή όχι και γιατί;

Είναι ο A* πλήρης;

Λύση: Τα κόστη των μεταβάσεων είναι πάντα θετικά, το b είναι πεπερασμένο και ξέρουμε ότι τότε ο A* είναι πλήρης, δηλαδή βρίσκει πάντα λύση αν υπάρχει.

Υπάρχει πάντα λύση;

Λύση: Εδώ υπάρχει πάντα λύση (υπάρχει πάντα ένα μονοπάτι μέχρι το Βουκουρέστι), επομένως βρίσκει πάντα λύση.

Είναι βέλτιστη;

Άσκηση 4.2(γ)

Μπορούμε να είμαστε σίγουροι ότι ο αλγόριθμος θα ανακαλύπτει πάντα το **συντομότερο μονοπάτι** προς το Βουκουρέστι, **από όποια πόλη του χάρτη και αν ξεκινήσουμε**; Ναι ή όχι και γιατί;

Είναι ο A* πλήρης;

Λύση: Τα κόστη των μεταβάσεων είναι πάντα θετικά, το b είναι πεπερασμένο και ξέρουμε ότι τότε ο A* είναι πλήρης, δηλαδή βρίσκει πάντα λύση αν υπάρχει.

Υπάρχει πάντα λύση;

Λύση: Εδώ υπάρχει πάντα λύση (υπάρχει πάντα ένα μονοπάτι μέχρι το Βουκουρέστι), επομένως βρίσκει πάντα λύση.

Είναι βέλτιστη;

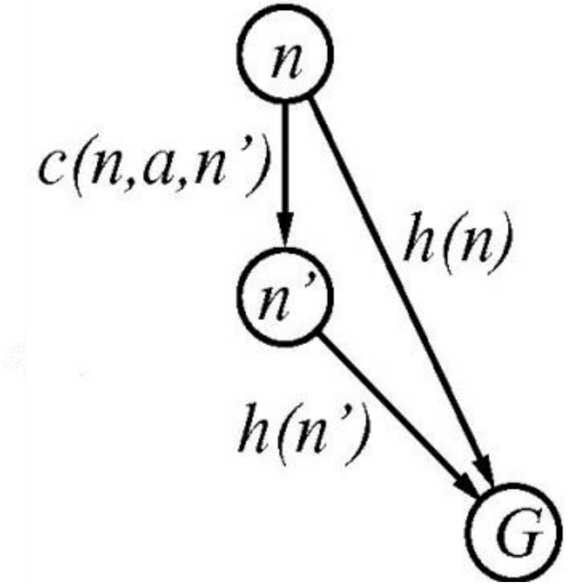
Λύση: η ευρετική είναι αποδεκτή και ξέρουμε ότι με αποδεκτή ευρετική ο A* είναι βέλτιστος.

Άσκηση 4.2(δ)

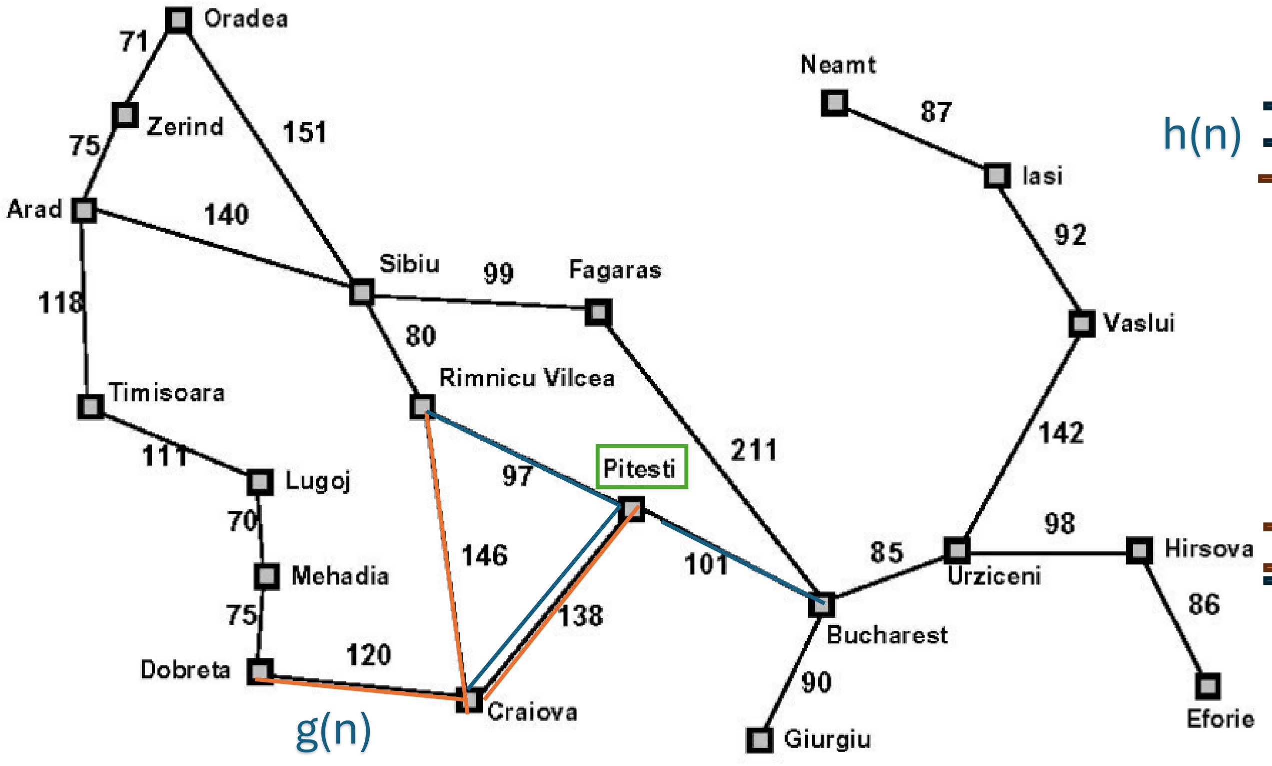
Χρησιμοποιούμε την ευθεία απόσταση μέχρι την πόλη στόχο

Είναι συνεπής η ευρετική που χρησιμοποιούμε; Ναι ή όχι και γιατί;

Συνεπής: **h(n) ≤ c(n, a, n') + h(n’)**



Άσκηση 4.2(δ)



h(n)

Straight−line distance to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Λύση: Ναι ισχύει h(n) ≤ c(n, n') + h(n’) όπου c η πραγματική οδική απόσταση

Άσκηση 4.2(ε)

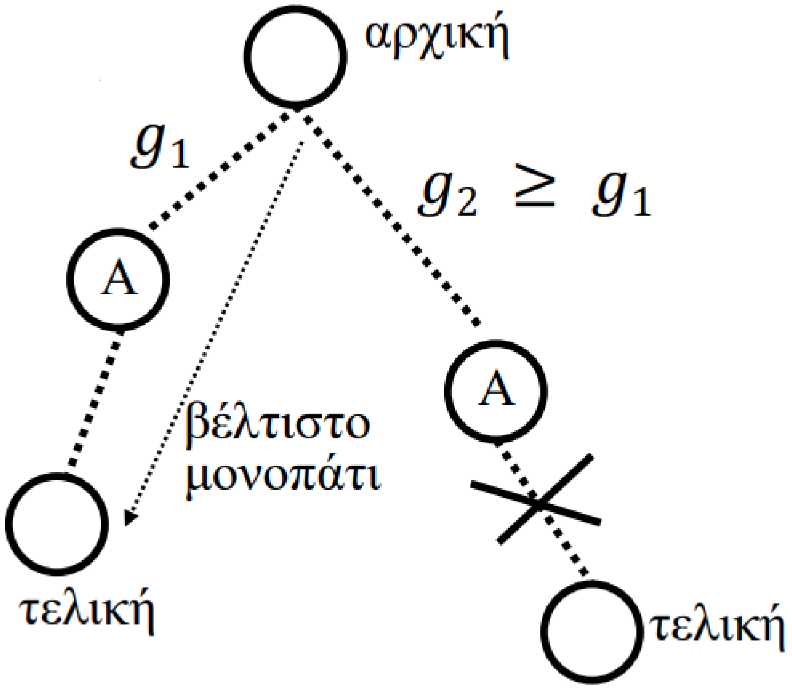
Αν προσθέσουμε **κλειστό σύνολο**, μπορούμε να είμαστε σίγουροι ότι ο αλγόριθμος θα ανακαλύπτει πάντα το **συντομότερο μονοπάτι** προς το Βουκουρέστι, **από όποια πόλη του χάρτη και αν ξεκινήσουμε**; Ναι ή όχι και γιατί;

Άσκηση 4.2(ε)

Λύση: Το κλειστό σύνολο ίσως μας αναγκάσει να διακόψουμε την εξερεύνηση ενός μονοπατιού κάτω από έναν κόμβο κατάστασης A την οποία έχουμε ξανασυναντήσει.

- Μήπως χάνουμε το βέλτιστο μονοπάτι;

Όχι, γιατί αφού η h είναι συνεπής, το προηγούμενο μονοπάτι με το οποίο είχαμε φτάσει σε A ήταν το συντομότερο μέχρι A.



Άσκηση 4.1(α)

Αποδείξτε ότι αν η h είναι συνεπής, τότε h(n_1) ≤ c(n → … → n_k) + h(n_k), για οποιοδήποτε μονοπάτι n_1 → … → n_k

Άσκηση 4.1(α)

Εφόσον η h είναι συνεπής θα ισχύει:

$\boxed{\text{h(n)} \leq \text{c(n} \rightarrow \text{n')} + \text{h(n') για κάθε n, n'.}}$

Άρα:

$h(n_1) \leq c(n_1 \rightarrow n_2) + h(n_2)$

$h(n_2) \leq c(n_2 \rightarrow n_3) + h(n_3)$

…

$h(n_{k-1}) \leq c(n_{k-1} \rightarrow n_k) + h(n_k)$

Και αθροίζοντας κατά μέλη έχουμε:

$h(n_1) \leq \sum_{i=2}^{k} c(n_{i-1} \rightarrow n_i) + h(n_k) \Rightarrow h(n_1) \leq c(n \rightarrow \ldots \rightarrow n_k) + h(n_k)$

Άσκηση 4.1(β)

Αποδείξτε ότι κάθε συνεπής h είναι και αποδεκτή.

Άσκηση 4.1(β)

Για να είναι αποδεκτή μια ευρετική, πρέπει να ισχύει ότι **h(n) ≤ C(n)*** για κάθε n. Έστω ότι η h(n) είναι συνεπής. Τότε, από το προηγούμενο σκέλος, χρησιμοποιώντας ως μονοπάτι (n = n_1) → … → n_k το βέλτιστο μονοπάτι από τον n ως κόμβο τελικής κατάστασης (όπου n_k ο κόμβος τελικής κατάστασης στην οποία τελειώνει το βέλτιστο μονοπάτι)

Εφόσον ο n_k είναι κόμβος τελικής κατάστασης, h(n_k) = 0.

Άρα, h(n) = h(n_1) ≤ c(n_1 → … → n_k) + h(n_k) = C(n)* + 0

Άσκηση 4.1(γ)

Αποδείξτε ότι αν οι $h_1, \ldots, h_k$ είναι αποδεκτές, τότε είναι αποδεκτή και η h(n) = max{h_1(n), …, h_k (n)}.

Άσκηση 4.1(γ)

Αφού όλες οι h_1(n), h_2(n), ... h_k(n) είναι αποδεκτές θα ισχύει:

h_1(n) ≤ C(n)*

h_2(n) ≤ C(n)*

...

h_k(n) ≤ C(n)*

Έχουμε ότι η h(n) = max{h_1(n), ..., h_k (n)} θα έχει σε κάθε n την τιμή μιας εκ των h_i(n) για 1 ≤ i ≤ k, οπότε h(n) = h_i(n) ≤ C(n)* και άρα h(n) αποδεκτή.

Άσκηση 4.1(δ)

Αποδείξτε ότι αν οι $h_1, \ldots, h_k$ είναι συνεπείς, τότε είναι συνεπής και η h(n) = max{h_1(n), …, h_k (n)}.

Άσκηση 4.1(δ)

Έστω ότι η h(n) = max{h_1(n), ..., h_k (n)} δεν είναι συνεπής, ενώ οι h_1(n), h_2(n), ... h_k(n) είναι συνεπείς. Τότε για κάποια n και n' :

$$h(n) > c(n \rightarrow n') + h(n') \quad (1)$$

Αφού η h(n) είναι μια εκ των h_1(n), h_2(n), ... h_k(n) τότε από (1):

$$\boxed{h(n) = h_i(n) > c(n \rightarrow n') + h(n') \quad (2)}$$

όπου 1 ≤ i ≤ k

Επειδή η h είναι συνεπής θα ισχύει:

$$h_i(n) \leq c(n \rightarrow n') + h(n') \quad (3)$$

Αφού όμως h(n') = max{h_1(n'), ..., h_k (n')} , τότε h_i(n') ≤ h_i(n') οπότε από (3) :

$$\boxed{h_i(n) \leq c(n \rightarrow n') + h(n') \quad (4)}$$

Άρα από (2) και (4) καταλήγουμε σε άτοπο.

Άσκηση 4.1(ε)

Εξηγήστε γιατί με ιδανική ευρετική συνάρτηση, η πολυπλοκότητα χρόνου του A* γίνεται O(bd).

Άσκηση 4.1(ε)

Λύση: Η ιδανική ευρετική μάς καθοδηγεί να επιλέγουμε και να **επεκτείνουμε κόμβους μόνο επί του βέλτιστου μονοπατιού**. Οι **κόμβοι αυτοί είναι d**, όσοι και το βάθος της ρηχότερης λύσης. Σε κάθε κόμβο κατά μήκος του βέλτιστου μονοπατιού παράγουμε όλα τα παιδιά του κόμβου, **δηλαδή b παιδιά στη χειρότερη περίπτωση**

Άσκηση 4.5(α)

1. Υπάρχει λύση
2. b πεπερασμένο
3. Σύνολο καταστάσεων πεπερασμένο
4. Κόστος λύσης αύξουσα συνάρτηση του βάθους (και μόνο)

ο Α* με **κλειστό σύνολο** και **αποδεκτή αλλά όχι συνεπή** ευρετική:

α) βρίσκει πάντα λύση και μάλιστα βέλτιστη

β) δεν βρίσκει πάντα λύση

γ) βρίσκει πάντα λύση, αλλά όχι σίγουρα βέλτιστη.

Άσκηση 4.5(α)

Λύση: Όταν χρησιμοποιείται **αποδεκτή ευρετική χωρίς κλειστό σύνολο**, ξέρουμε επίσης ότι ο **A* είναι βέλτιστος**. Με κλειστό σύνολο, όμως, αν η ευρετική είναι απλά αποδεκτή και όχι συνεπής, τότε ο A* δεν επιστρέφει σίγουρα τη βέλτιστη λύση, γιατί **ενδέχεται το μονοπάτι της βέλτιστης λύσης να πριονίζεται λόγω της χρήσης του κλειστού συνόλου.**

Άσκηση 4.5(β)

1. Υπάρχει λύση
2. b πεπερασμένο
3. Σύνολο καταστάσεων πεπερασμένο
4. Κόστος λύσης αύξουσα συνάρτηση του βάθους (και μόνο)

ο A* με **κλειστό σύνολο** και **συνεπή** ευρετική:

α) βρίσκει πάντα λύση και μάλιστα βέλτιστη

β) δεν βρίσκει πάντα λύση

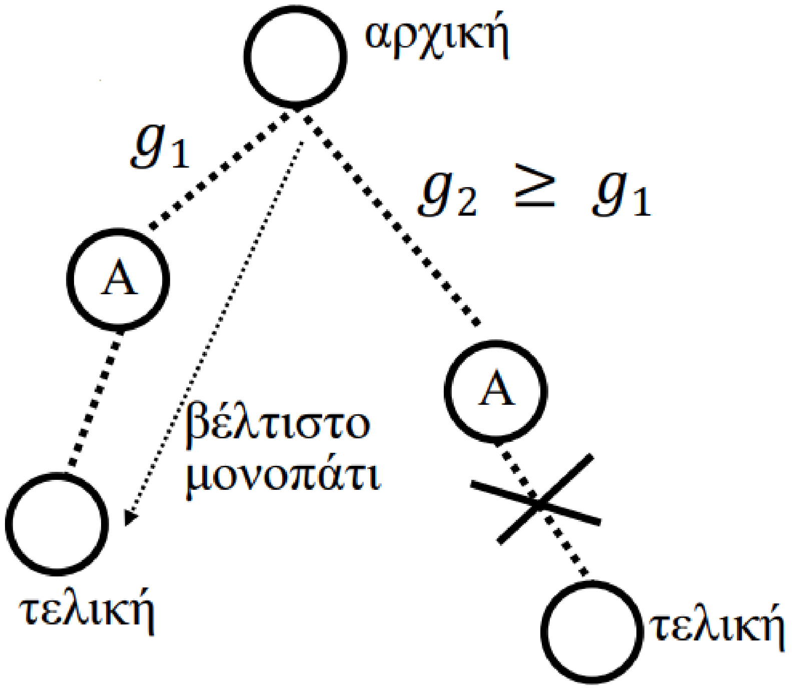
γ) βρίσκει πάντα λύση, αλλά όχι σίγουρα βέλτιστη.

Άσκηση 4.5(β)

Λύση: Το κλειστό σύνολο ίσως μας αναγκάσει να διακόψουμε την εξερεύνηση ενός μονοπατιού κάτω από έναν κόμβο κατάστασης A την οποία έχουμε ξανασυναντήσει.

- Μήπως χάνουμε το βέλτιστο μονοπάτι;

Όχι, γιατί αφού η h είναι συνεπής, το προηγούμενο μονοπάτι με το οποίο είχαμε φτάσει σε A ήταν το συντομότερο μέχρι A.



Άσκηση 4.5(β)

1. Υπάρχει λύση
2. b πεπερασμένο
3. Σύνολο καταστάσεων πεπερασμένο
4. Κόστος λύσης αύξουσα συνάρτηση του βάθους (και μόνο)

ο A* με **κλειστό σύνολο** και **συνεπή** ευρετική:

<u>α) βρίσκει πάντα λύση και μάλιστα βέλτιστη</u>

β) δεν βρίσκει πάντα λύση

γ) βρίσκει πάντα λύση, αλλά όχι σίγουρα βέλτιστη.

Άσκηση 4.5(γ)

1. **ΔΕΝ** υπάρχει λύση
2. b πεπερασμένο
3. Σύνολο καταστάσεων πεπερασμένο

ο αλγόριθμος A* **με κλειστό σύνολο** και **μη αποδεκτή ευρετική:**

α) τερματίζει πάντα

β) δεν τερματίζει ποτέ

γ) άλλοτε τερματίζει και άλλοτε δεν τερματίζει

Άσκηση 4.5(γ)

Λύση: το δέντρο αναζήτησης που θα ψάξει ο A* είναι πεπερασμένο. Αφού δεν υπάρχει λύση, ο A* θα το ψάξει ολόκληρο και θα τερματίσει.

Το ότι η ευρετική είναι μη αποδεκτή δεν παίζει ρόλο εδώ.

Άσκηση 4.5(γ)

1. **ΔΕΝ** υπάρχει λύση
2. b πεπερασμένο
3. Σύνολο καταστάσεων πεπερασμένο

ο αλγόριθμος Α* **με κλειστό σύνολο** και **μη αποδεκτή ευρετική:**

<u>α) τερματίζει πάντα</u>

β) δεν τερματίζει ποτέ

γ) άλλοτε τερματίζει και άλλοτε δεν τερματίζει

Άσκηση 4.5(δ)

Ο Α* χωρίς κλειστό σύνολο έχει:

α) καλύτερη

β) χειρότερη

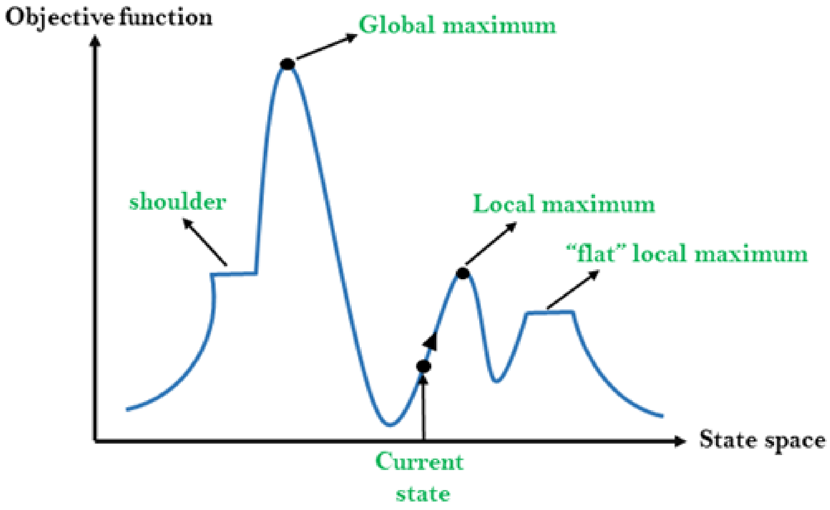
γ) την ίδια

πολυπλοκότητα **χώρου** από τον **αλγόριθμο αναρρίχησης λόφου** χωρίς **κλειστό σύνολο**

Άσκηση 4.5(δ)

Λύση: Χωρίς κλειστό σύνολο, ο αλγόριθμος αναρρίχησης λόφου κρατά σε κάθε βήμα του στη μνήμη **το πολύ την τρέχουσα κατάσταση και τα b παιδιά της**

Αντιθέτως, ο A* κρατά στη μνήμη εν γένει πολλούς κόμβους του μετώπου της αναζήτησης (και τους προγόνους τους, αν θέλουμε να μας επιστρέψει το μονοπάτι της λύσης).



Άσκηση 4.5(δ)

Ο A* χωρίς κλειστό σύνολο έχει:

α) καλύτερη

<u>β) χειρότερη</u>

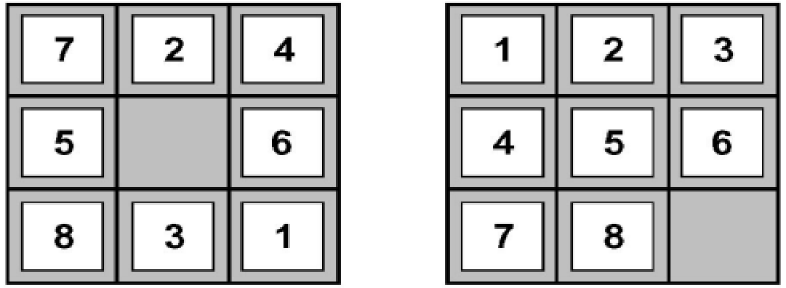
γ) την ίδια

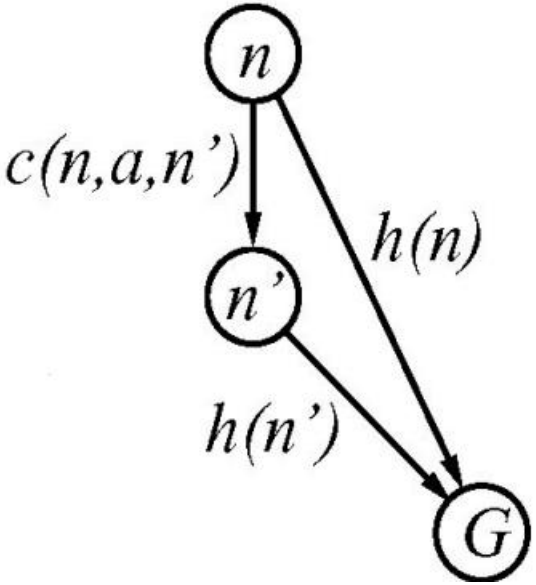
πολυπλοκότητα **χώρου** από τον **αλγόριθμο αναρρίχησης λόφου** χωρίς **κλειστό σύνολο**

Άσκηση 4.6

Αποδείξτε ότι στο πρόβλημα των πλακιδίων η ευρετική που μετρά τον αριθμό πλακιδίων εκτός θέσης είναι **συνεπής**

Συνεπής: $\mathbf{h(n) \leq c(n, a, n') + h(n')}$





Άσκηση 4.6

Έχουμε 3 περιπτώσεις όταν κάνουμε μια κίνηση:

- Η κίνηση μετακινεί στη **σωστή θέση** ένα πλακίδιο που βρισκόταν σε **λανθασμένη θέση.**
- Η κίνηση μετακινεί σε **λανθασμένη θέση** ένα πλακίδιο που βρισκόταν στη **σωστή θέση.**
- Η κίνηση μετακινεί ένα πλακίδιο από **λανθασμένη θέση** σε **λανθασμένη θέση**.

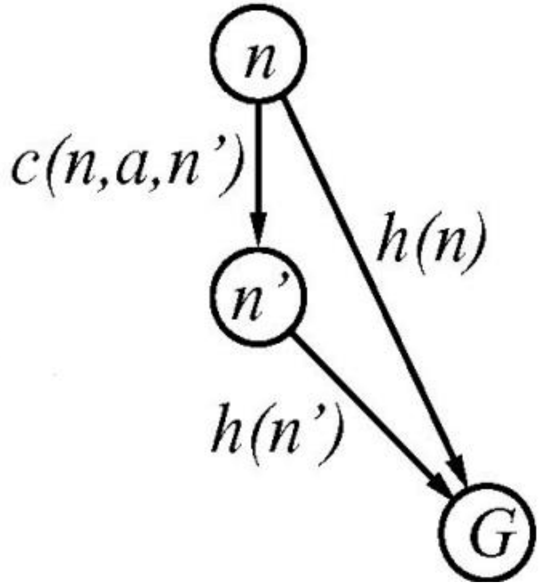
Άσκηση 4.6

- Η κίνηση μετακινεί στη **σωστή θέση** ένα πλακίδιο που βρισκόταν σε **λανθασμένη θέση.**

$h(n') = h(n) – 1$

Ένα πλακίδιο μπήκε σε σωστή θέση,

άρα ένα λιγότερο εκτός θέσεως.



$c(n, a, n') + h(n') = 1 + h(n) – 1 = h(n)$

Άρα, $\mathbf{h(n) \leq c(n, a, n') + h(n')}$

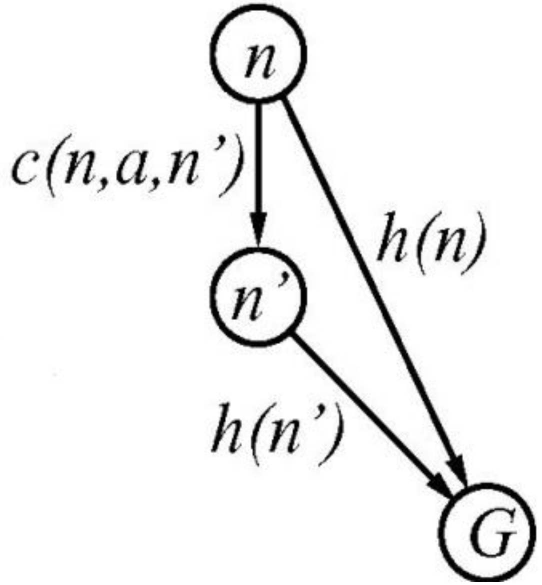
Άσκηση 4.6

- Η κίνηση μετακινεί στη **λανθασμένη θέση** ένα πλακίδιο που βρισκόταν σε **σωστή θέση.**

$h(n') = h(n) + 1$

Ένα πλακίδιο μπήκε σε λανθασμένη θέση,

άρα ένα ακόμη εκτός θέσεως.



$c(n, a, n') + h(n') = 1 + h(n) + 1 = h(n) + 2$

Άρα, $\mathbf{h(n) \leq c(n, a, n') + h(n')}$

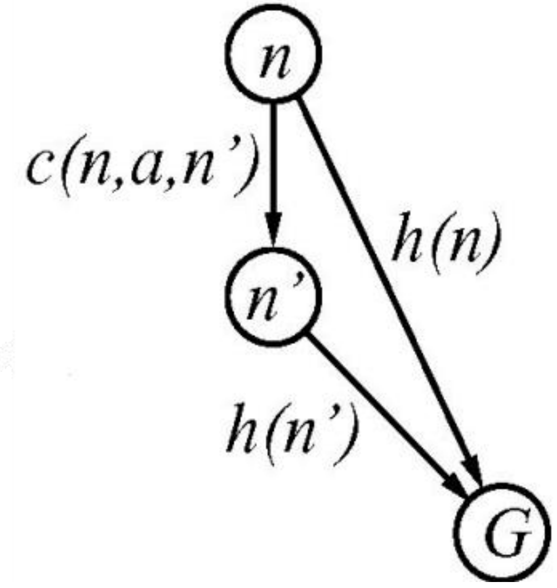
Άσκηση 4.6

- Η κίνηση μετακινεί ένα πλακίδιο από **λανθασμένη θέση** σε **λανθασμένη θέση**.

h(n’) = h(n)

c(n, a, n') + h(n’) = 1 + h(n)

Άρα, **h(n) ≤ c(n, a, n') + h(n’)**



Άσκηση 4.7

Αποδείξτε ότι οι ευρετικές που προκύπτουν με **αφαίρεση περιορισμών** είναι **συνεπείς**, αν τα **κόστη** των μεταβάσεων είναι πάντα **≥ 0** (και ίδια στο αρχικό και στο χαλαρωμένο πρόβλημα).

Αφαίρεση περιορισμών: πχ αν μπορούν να τοποθετηθούν πολλά πλακίδια στην ίδια θέση

Άσκηση 4.7

Η ευρετική h(n) του αρχικού προβλήματος είναι το **ακριβές βέλτιστο κόστος λύσης του απλοποιημένου** προβλήματος.

h(n) ≤ c(n, a, n') + h(n')

Οι δυνατές μεταβάσεις του **αρχικού προβλήματος** είναι **υποσύνολο των δυνατών μεταβάσεων του απλοποιημένου** (και τα κόστη των μεταβάσεων είναι ίδια και **≥ 0**). Άρα ικανοποιείται η τριγωνική ανισότητα, οπότε η h(n) είναι συνεπής.