

ΠΛΗ31

ΕΝΟΤΗΤΑ 1: ΑΝΑΖΗΤΗΣΗ

Μάθημα 1.2:
Τυφλή Αναζήτηση

Δημήτρης Ψούνης



www.psounis.gr

ΠΕΡΙΕΧΟΜΕΝΑ

A. Σκοπός του Μαθήματος

B. Θεωρία

1. Εισαγωγή

1. Ορισμοί για Δένδρα
2. Χώρος Καταστάσεων – Χώρος Αναζήτησης

2. Αλγόριθμοι Τυφλής Αναζήτησης

1. Αναζήτηση Κατά Βάθος

1. Ψευδογλώσσα
2. Παράδειγμα Εκτέλεσης
3. Παρατηρήσεις
4. Βελτιστότητα
5. Πληρότητα
6. Πολυπλοκότητα Χρόνου
7. Πολυπλοκότητα Χώρου

2. Αναζήτηση Κατά Πλάτος

1. Ψευδογλώσσα
2. Παράδειγμα Εκτέλεσης
3. Παρατηρήσεις
4. Βελτιστότητα
5. Πληρότητα
6. Πολυπλοκότητα Χρόνου
7. Πολυπλοκότητα Χώρου

Γ. Ασκήσεις

A. Σκοπός του Μαθήματος

Επίπεδο A

- Οι αλγόριθμοι τυφλής αναζήτησης Κατά Βάθος και Κατά Πλάτος

Επίπεδο B

- Πληρότητα, Βελτιστότητα, Πολυπλοκότητα Χρόνου και Πολυπλοκότητα Χώρου των αλγορίθμων τυφλής αναζήτησης.

- Το Πρόβλημα του Λαβυρίνθου

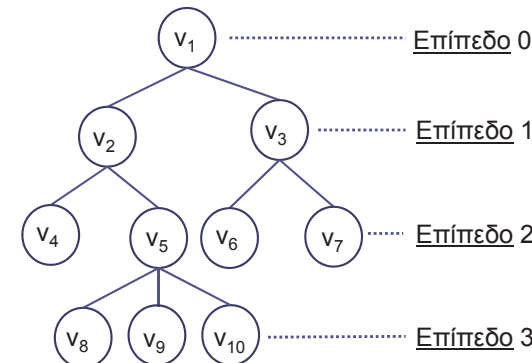
Επίπεδο Γ

- (-)

B. Θεωρία

1. Εισαγωγή

1. Ορισμοί Δένδρων



Ρίζα του δένδρου (είναι μοναδική) Στο παράδειγμα η v_1
Φύλλα (= κόμβοι που δεν έχουν παιδιά). Στο π.χ.: $v_4, v_6, v_7, v_8, v_9, v_{10}$
Εσωτερικοί Κόμβοι (= κόμβοι που έχουν παιδιά). Στο π.χ.: v_1, v_2, v_3, v_5

Για έναν συγκεκριμένο κόμβο (π.χ. για τον v_5)

- Πατέρας του κόμβου είναι ο v_2
- Πρόγονοι του κόμβου είναι οι v_2 και v_1
- Παιδιά του κόμβου είναι οι v_8, v_9, v_{10}

Μονοπάτι από την ρίζα έως το v_9
 Είναι το $v_1-v_2-v_5-v_9$

ΒΑΘΟΣ ΔΕΝΔΡΟΥ = Μέγιστο Επίπεδο (Στο π.χ.=3) (Συμβολίζεται συνήθως με d)

ΠΑΡΑΓΟΝΤΑΣ ΔΙΑΚΛΑΔΩΣΗΣ = Μέγιστο Πλήθος Παιδιών που έχει ένας εσωτερικός κόμβος (Στο π.χ. 3). (Συμβολίζεται συνήθως με b)
 Το δένδρο λέγεται b-αδικό

B. Θεωρία

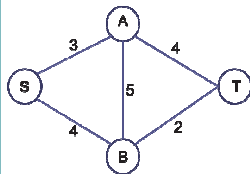
1. Εισαγωγή

2. Γράφος Αναζήτησης-Χώρος Αναζήτησης

Από το προηγούμενο μάθημα υπενθυμίζουμε:

ΧΩΡΟΣ ΚΑΤΑΣΤΑΣΕΩΝ (ή **ΓΡΑΦΟΣ ΚΑΤΑΣΤΑΣΕΩΝ**) είναι ένας γράφος που έχει:

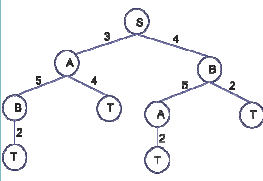
- ως κορυφές όλες τις έγκυρες καταστάσεις
- ως ακμές (κατευθυνόμενες ή μη κατευθυνόμενες ανάλογα με το πρόβλημα) τις ενέργειες που συνδέουν μια κατάσταση με μια άλλη.



ΧΩΡΟΣ ΑΝΑΖΗΤΗΣΗΣ είναι ένας δένδρο που περιέχει όλα τα μονοπάτια από την αφετηρία προς τον κόμβο-στόχο

Η παραγωγή του γίνεται με συγκεκριμένο αλγόριθμο: Εξετάζουμε όλους του κόμβους και:

- Θέτουμε ως παιδιά του κόμβου τους γείτονες στον γράφο, που δεν είναι πρόγονοι στο δένδρο.

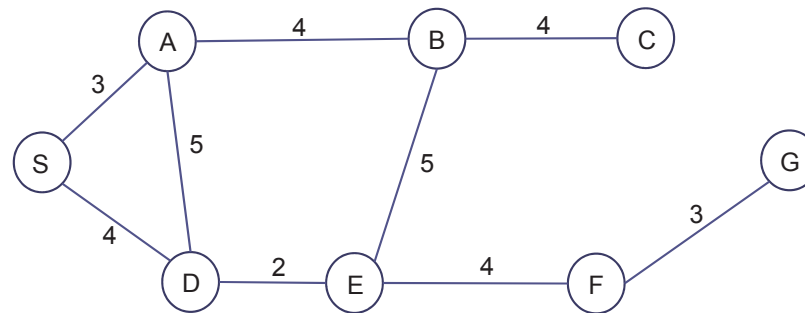


B. Θεωρία

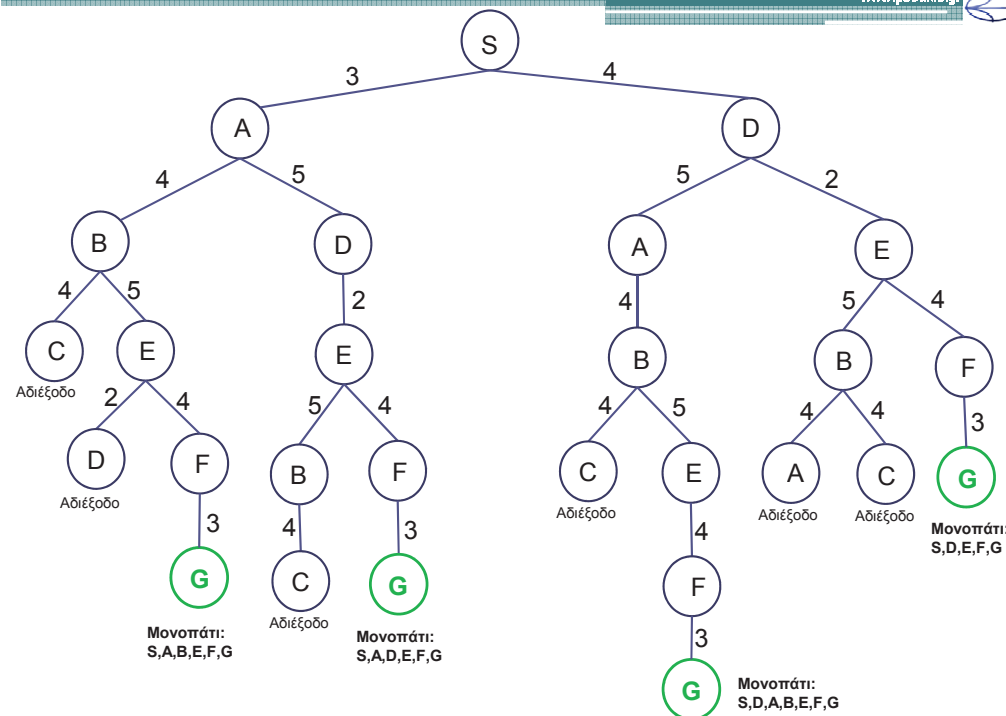
1. Εισαγωγή

1. Χώρος Αναζήτησης

Βλέπουμε ένα ακόμη παράδειγμα ενός χώρου καταστάσεων:



- Και έστω ότι αρχική κατάσταση είναι η S και τελική κατάσταση είναι η G. Στην επόμενη διαφάνεια φαίνεται ο χώρος αναζήτησης



B. Θεωρία

1. Εισαγωγή

2. Δένδρο Αναζήτησης

- Ο χώρος αναζήτησης είναι ΤΕΡΑΣΤΙΟΣ και δεν μπορεί να εξερευνηθεί εξ' ολοκλήρου, ιδίως για μεγάλους χώρους καταστάσεων
- Οι αλγόριθμοι αναζήτησης εξερευνούν τον χώρο αναζήτησης ως εξής:
- Με μία μεθοδολογία (που εξαρτάται από τον αλγόριθμο) αναπτύσσουν έναν κόμβο μόνο εφόσον φαίνεται υποσχόμενος ότι θα οδηγήσει σε μία λύση.
 - Έτσι σταδιακά, κατασκευάζουν ένα μέρος του χώρου αναζήτησης ο οποίος θα αποτελεί το ΔΕΝΔΡΟ ΑΝΑΖΗΤΗΣΗΣ του συγκεκριμένου αλγορίθμου.

ΔΕΝΔΡΟ ΑΝΑΖΗΤΗΣΗΣ είναι το μέρος του χώρου αναζήτησης που παράγεται από έναν αλγόριθμο αναζήτησης



B. Θεωρία

2. Αλγόριθμοι Τυφλής Αναζήτησης

- Για την εύρεση ενός μονοπατιού-λύσης έχουμε τους μηχανισμούς πλοήγησης που χωρίζονται σε δύο κατηγορίες:
 - Αλγόριθμοι Τυφλής Αναζήτησης (είναι αλγόριθμοι που αναζητούν μια λύση χωρίς να λαμβάνουν υπόψιν τα βάρη των ακμών)
 - Αλγόριθμος αναζήτησης κατά βάθος
 - Αλγόριθμος αναζήτησης κατά πλάτος
 - Αλγόριθμοι Ευρετικής Αναζήτησης (που λαμβάνουν υπόψιν τα βάρη των ακμών)
 - Απληστος Αλγόριθμος
 - Αλγόριθμος A*
 - Αλγόριθμος UCS

B. Θεωρία

2. Αλγόριθμοι Τυφλής Αναζήτησης

1. Αλγόριθμος Αναζήτησης Κατά Βάθος (1.Ψευδογλώσσα)

- Ο αλγόριθμος διάσχισης κατά βάθος κινείται σαν **βολίδα** κάνοντας μια βουτιά στον χώρο αναζήτησης προσπαθώντας να εντοπίσει μία λύση:

ΑΛΓΟΡΙΘΜΟΣ ΚΑΤΑ ΒΑΘΟΣ

Βήμα 1: Δημιούργησε το δένδρο αναζήτησης T, το οποίο αρχικά αποτελείται από τον κόμβο αρχής n_0

Βήμα 1.1: Δημιούργησε μια **στοίβα** και ονόμασε την **ΑΝΟΙΚΤΕΣ**

Βήμα 1.2: Δημιούργησε μια **λίστα** και ονόμασε την **ΚΛΕΙΣΤΕΣ**

Βήμα 1.3: Τοποθέτησε τον n_0 στην ΑΝΟΙΚΤΕΣ

Βήμα 2: Αν η ΑΝΟΙΚΤΕΣ είναι κενή στοίβα τότε τερμάτισε με αποτυχία

Βήμα 2.1: Επέλεξε το πρώτο στοιχείο της και αφάιρεσε το από αυτήν

Βήμα 2.2: Ονόμασε αυτό το στοιχείο n

Βήμα 3: Αν το n είναι ο στόχος τότε τερμάτισε με επιτυχία και επέστρεψε τη λύση.

Βήμα 3.1: **Επέκτεινε** το n , δηλαδή εμφάνισε τα παιδιά της που δεν είναι στις κλειστές

Βήμα 3.2: Τοποθέτησε τους παραγόμενους κόμβους στην αρχή της στοίβας.

Βήμα 3.3: Βάλε το n στις ΚΛΕΙΣΤΕΣ και αφάιρεσε όλες τις εμφανίσεις του n από τις ΑΝΟΙΚΤΕΣ

Βήμα 4: Επέστρεψε στο βήμα 2.

B. Θεωρία

2. Αλγόριθμοι Τυφλής Αναζήτησης

1. Αλγόριθμος Αναζήτησης Κατά Βάθος (1.Ψευδογλώσσα)

Εμπειρικά:

ΑΛΓΟΡΙΘΜΟΣ ΚΑΤΑ ΒΑΘΟΣ

Αρχικά:

- Βάζουμε την αφετηρία στο δένδρο.

Επαναληπτικά:

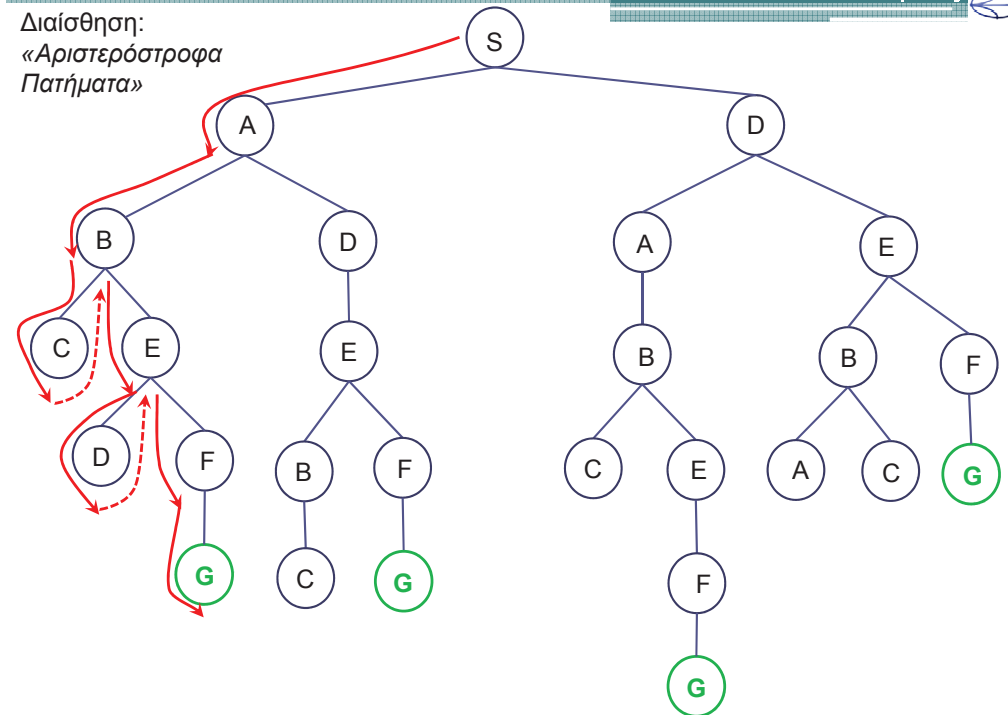
- **Πατάμε** «αριστερόστροφα» στον επόμενο κόμβο.
- **Ανοίγουμε** τους γείτονες του (που δεν είναι πρόγονοί του) και τους θέτουμε ως παιδιά του
- **Διαγράφουμε** ανοικτές εμφανίσεις των κόμβων που έχουμε πατήσει.

Εως ότου:

- Πατήσουμε στον κόμβο-στόχο

Διαίσθηση:

«Αριστερόστροφα
Πατήματα»



Β. Θεωρία

2. Αλγόριθμοι Τυφλής Αναζήτησης

1. Αλγόριθμος Αναζήτησης Κατά Βάθος (2.Παράδειγμα Εκτέλεσης)

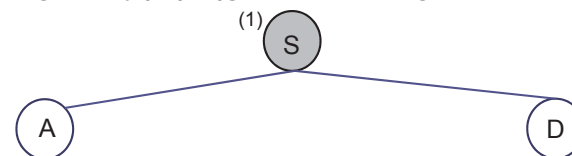


Βήμα	ΑΝΟΙΚΤΕΣ	ΚΛΕΙΣΤΕΣ
0	(S)	{}

Β. Θεωρία

2. Αλγόριθμοι Τυφλής Αναζήτησης

1. Αλγόριθμος Αναζήτησης Κατά Βάθος (2.Παράδειγμα Εκτέλεσης)

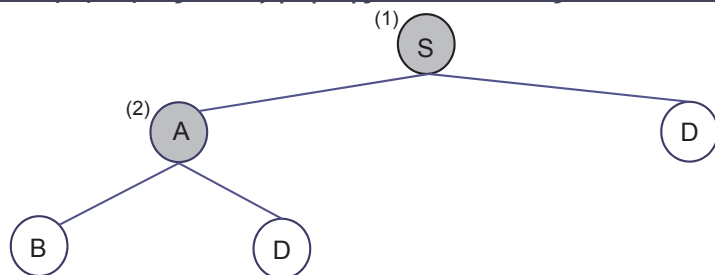


Βήμα	ΑΝΟΙΚΤΕΣ	ΚΛΕΙΣΤΕΣ
0	(S)	{}
1	(A,D)	{S}

Β. Θεωρία

2. Αλγόριθμοι Τυφλής Αναζήτησης

1. Αλγόριθμος Αναζήτησης Κατά Βάθος (2.Παράδειγμα Εκτέλεσης)

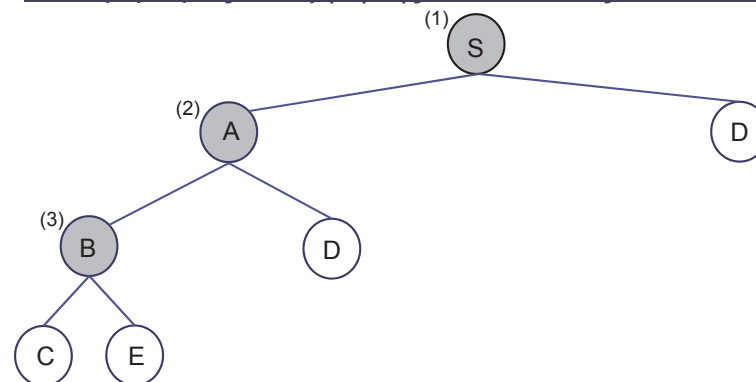


Βήμα	ΑΝΟΙΚΤΕΣ	ΚΛΕΙΣΤΕΣ
0	(S)	{}
1	(A,D)	{S}
2	(B,D _{SA} ,D _S)	{S,A}

Β. Θεωρία

2. Αλγόριθμοι Τυφλής Αναζήτησης

1. Αλγόριθμος Αναζήτησης Κατά Βάθος (2.Παράδειγμα Εκτέλεσης)

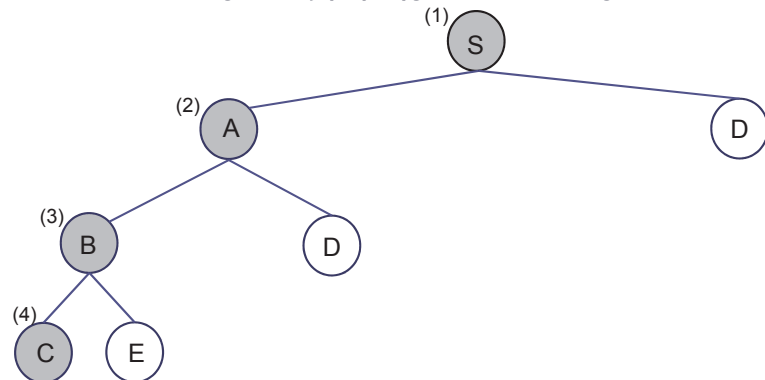


Βήμα	ΑΝΟΙΚΤΕΣ	ΚΛΕΙΣΤΕΣ
0	(S)	{}
1	(A,D)	{S}
2	(B,D _{SA} ,D _S)	{S,A}
3	(C,E,D _{SA} ,D _S)	{S,A,B}

Β. Θεωρία

2. Αλγόριθμοι Τυφλής Αναζήτησης

1. Αλγόριθμος Αναζήτησης Κατά Βάθος (2.Παράδειγμα Εκτέλεσης)

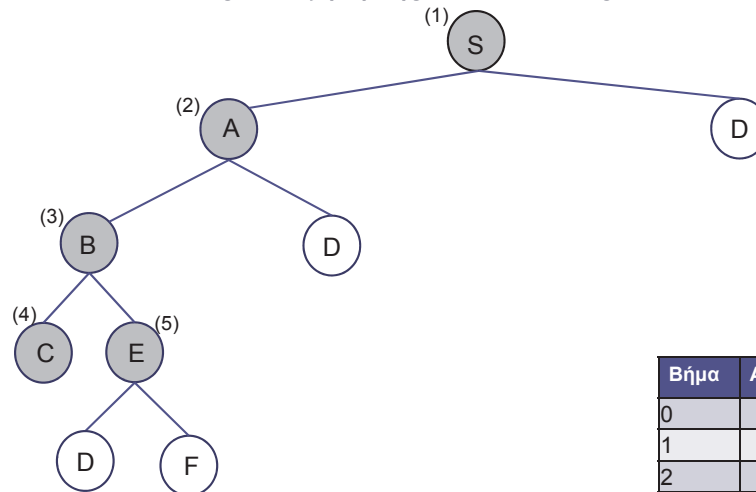


Βήμα	ΑΝΟΙΚΤΕΣ	ΚΛΕΙΣΤΕΣ
0	(S) {}	
1	(A,D) {S}	
2	(B, D _{SA} , D _S) {S,A}	
3	(C,E, D _{SA} , D _S) {S,A,B}	
4	(E, D _{SA} , D _S) {S,A,B,C}	

Β. Θεωρία

2. Αλγόριθμοι Τυφλής Αναζήτησης

1. Αλγόριθμος Αναζήτησης Κατά Βάθος (2.Παράδειγμα Εκτέλεσης)

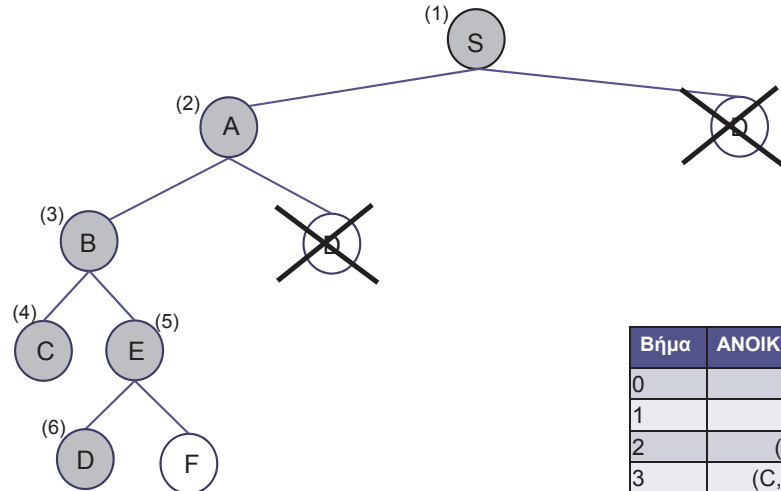


Βήμα	ΑΝΟΙΚΤΕΣ	ΚΛΕΙΣΤΕΣ
0	(S) {}	
1	(A,D) {S}	
2	(B, D _{SA} , D _S) {S,A}	
3	(C,E, D _{SA} , D _S) {S,A,B}	
4	(E, D _{SA} , D _S) {S,A,B,C}	
5	(D _{SABE} , F, D _{SA} , D _S) {S,A,B,C,E}	

Β. Θεωρία

2. Αλγόριθμοι Τυφλής Αναζήτησης

1. Αλγόριθμος Αναζήτησης Κατά Βάθος (2.Παράδειγμα Εκτέλεσης)

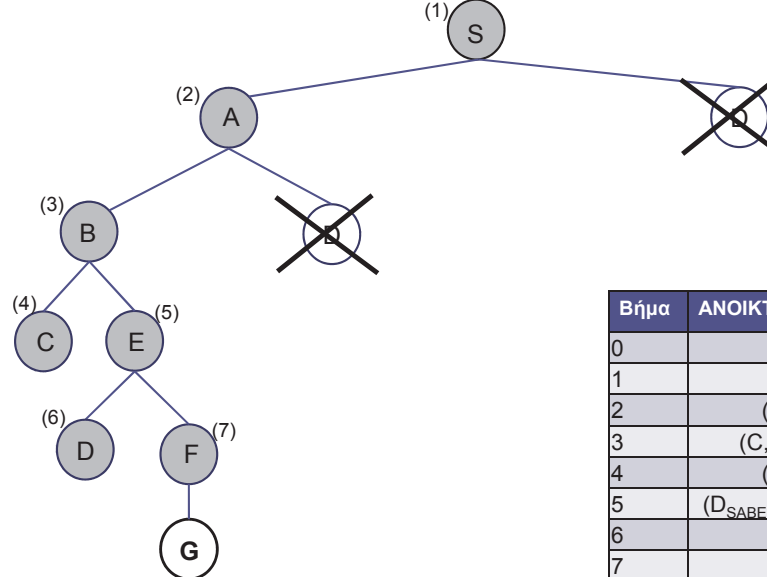


Βήμα	ΑΝΟΙΚΤΕΣ	ΚΛΕΙΣΤΕΣ
0	(S) {}	
1	(A,D) {S}	
2	(B, D _{SA} , D _S) {S,A}	
3	(C,E, D _{SA} , D _S) {S,A,B}	
4	(E, D _{SA} , D _S) {S,A,B,C}	
5	(D _{SABE} , F, D _{SA} , D _S) {S,A,B,C,E}	
6	(F) {S,A,B,C,E,D}	

Β. Θεωρία

2. Αλγόριθμοι Τυφλής Αναζήτησης

1. Αλγόριθμος Αναζήτησης Κατά Βάθος (2.Παράδειγμα Εκτέλεσης)

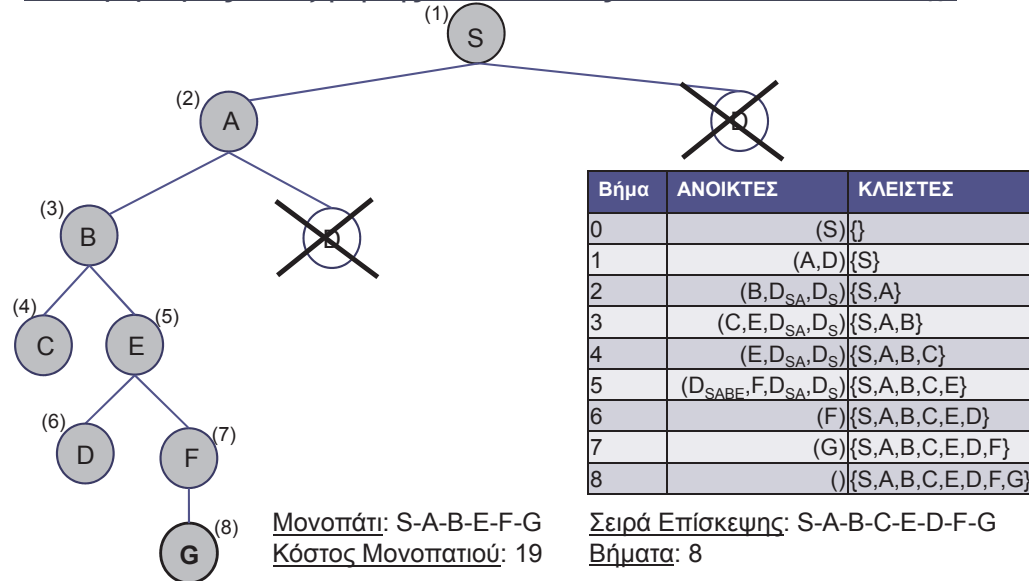


Βήμα	ΑΝΟΙΚΤΕΣ	ΚΛΕΙΣΤΕΣ
0	(S) {}	
1	(A,D) {S}	
2	(B, D _{SA} , D _S) {S,A}	
3	(C,E, D _{SA} , D _S) {S,A,B}	
4	(E, D _{SA} , D _S) {S,A,B,C}	
5	(D _{SABE} , F, D _{SA} , D _S) {S,A,B,C,E}	
6	(F) {S,A,B,C,E,D}	
7	(G) {S,A,B,C,E,D,F}	

B. Θεωρία

2. Αλγόριθμοι Τυφλής Αναζήτησης

1. Αλγόριθμος Αναζήτησης Κατά Βάθος (2.Παράδειγμα Εκτέλεσης)



B. Θεωρία

2. Αλγόριθμοι Τυφλής Αναζήτησης

1. Αλγόριθμος Αναζήτησης Κατά Βάθος (3.Παρατηρήσεις)

ΠΑΡΑΤΗΡΗΣΗ 1: Σειρά Τοποθέτησης των Παιδιών

- Υιοθετούμε την εξής σύμβαση: Όταν εισάγουμε τα παιδιά ενός κόμβου η σειρά με την οποία τα βάζουμε στην λίστα ΑΝΟΙΚΤΕΣ είναι λεξικογραφική (αλφαβητική)
- Αν ακολουθήσουμε διαφορετική σύμβαση θα προκύψει διαφορετική εκτέλεση της κατά βάθος που ίσως οδηγήσει σε άλλη λύση
 - Αν δεν καθορίζεται από την εκφώνηση ακολουθούμε την λεξικογραφική σειρά και θα αναφέρουμε ότι κάνουμε την συγκεκριμένη παραδοχή.
 - Σε κάποιες εκφωνήσεις θα μας δίνεται η σειρά των κόμβων για την διάσπαση των ισοπαλιών, δηλαδή για την απόφαση με ποια σειρά εισάγονται οι κόμβοι στην λίστα ΑΝΟΙΚΤΕΣ

B. Θεωρία

2. Αλγόριθμοι Τυφλής Αναζήτησης

1. Αλγόριθμος Αναζήτησης Κατά Βάθος (3.Παρατηρήσεις)

ΠΑΡΑΤΗΡΗΣΗ 2: Παραλλαγές του Αλγορίθμου Κατά Βάθος

- Το ΕΑΠ δεν έχει διαλευκάνει το κριτήριο τερματισμού του αλγορίθμου
- Συγκεκριμένα ακολουθούνται δύο κριτήρια τερματισμού:
 - Να πατήσουμε στον κόμβο-στόχο (όπως στις διαφάνειες αυτές)
 - Οπότε και σταματάμε στο βήμα 8
 - Να εμφανιστεί ο κόμβος-στοχος στους απογόνους του κόμβου που «πατάμε»
 - Στο παράδειγμα μας θα σταματούσαμε στο βήμα 7.

B. Θεωρία

2. Αλγόριθμοι Τυφλής Αναζήτησης

1. Αλγόριθμος Αναζήτησης Κατά Βάθος (4.Πληρότητα)

ΠΛΗΡΟΤΗΤΑ ΑΛΓΟΡΙΘΜΟΥ ΑΝΑΖΗΤΗΣΗΣ

Ένας αλγόριθμος αναζήτησης θα λέμε ότι είναι πλήρης, εάν σίγουρα μπορεί να εντοπίσει μια (οποιαδήποτε) λύση στον χώρο αναζήτησης εφόσον αυτή υπάρχει. Εάν δεν υπάρχει θα πρέπει να είναι σε θέση να απαντήσει ότι δεν υπάρχει λύση.

- Ο αλγόριθμος αναζήτησης Κατά Βάθος ΔΕΝ ΕΙΝΑΙ πλήρης
 - Σε πεπερασμένους χώρους αναζήτησης είναι πλήρης.
 - Σε άπειρους χώρους αναζήτησης μπορεί να εγκλωβιστεί σε υποδένδρο που δεν περιέχει λύση του προβλήματος, άρα δεν θα τερματίζει ποτέ.

Β. Θεωρία

2. Αλγόριθμοι Τυφλής Αναζήτησης

1. Αλγόριθμος Αναζήτησης Κατά Βάθος (5.Βέλτιστος)

Πότε ένας ΑΛΓΟΡΙΘΜΟΣ ΑΝΑΖΗΤΗΣΗΣ είναι ΒΕΛΤΙΣΤΟΣ

Ένας αλγόριθμος αναζήτησης θα λέμε ότι είναι ΒΕΛΤΙΣΤΟΣ με βάση δύο κριτήρια:

A: Αν επιστρέφει την λύση που είναι κοντινότερα στην ρίζα (Αν ο Γράφος δεν έχει Βάρη).

B: Αν επιστρέφει την λύση που έχει το μικρότερο κόστος (Αν ο Γράφος έχει Βάρη).

➤ Ο αλγόριθμος αναζήτησης Κατά Βάθος ΔΕΝ ΕΙΝΑΙ ΒΕΛΤΙΣΤΟΣ

- Όσον αφορά την απόσταση από την ρίζα: Εντοπίζει την πρώτη λύση κατά τη διάρκεια της «βουτιάς» ενώ μπορεί σε άλλο κλάδο να βρίσκεται μια λύση πιο κοντά στην ρίζα
- Όσον αφορά το κόστος: Δεν συνεκτιμά τα κόστη των ακμών, άρα δεν επιστρέφει την βέλτιστη λύση.

Β. Θεωρία

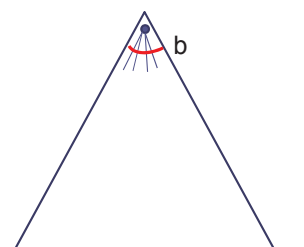
2. Αλγόριθμοι Τυφλής Αναζήτησης

1. Αλγόριθμος Αναζήτησης Κατά Βάθος (6. Πολ/τα Χρόνου)

ΧΡΟΝΙΚΗ ΠΟΛΥΠΛΟΚΟΤΗΤΑ ενός ΑΛΓΟΡΙΘΜΟΥ ΑΝΑΖΗΤΗΣΗΣ

Χρονική πολυπλοκότητα ενός αλγόριθμου αναζήτησης είναι το πλήθος των κόμβων του χώρου αναζήτησης που απαιτείται να εξετάσει ο αλγόριθμος προκειμένου να οδηγηθεί σε μία λύση.

- Κάνουμε μια ανάλυση χειρότερης περίπτωσης σε ένα δένδρο που έχει βάθος d και έχει παράγοντα διακλάδωσης b :



Βάθος: d

Στην χειρότερη περίπτωση ο αλγόριθμος θα εξετάσει όλους τους κόμβους του δένδρου:

$$1 + b + b^2 + \dots + b^d = \frac{b^{d+1} - 1}{b - 1} = O(b^{d+1})$$

Άρα μιλάμε για εκθετική πολυπλοκότητα

Β. Θεωρία

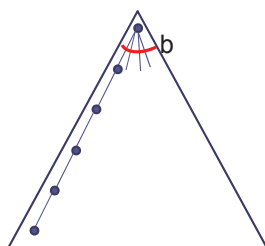
2. Αλγόριθμοι Τυφλής Αναζήτησης

1. Αλγόριθμος Αναζήτησης Κατά Βάθος (7. Πολ/τα Χώρου)

ΧΩΡΙΚΗ ΠΟΛΥΠΛΟΚΟΤΗΤΑ ενός ΑΛΓΟΡΙΘΜΟΥ ΑΝΑΖΗΤΗΣΗΣ

Χωρική πολυπλοκότητα ενός αλγόριθμου αναζήτησης είναι το πλήθος των κόμβων του χώρου αναζήτησης που θα πρέπει να αποθηκεύονται στην λίστα ΑΝΟΙΚΤΕΣ.

- Κάνουμε μια ανάλυση χειρότερης περίπτωσης σε ένα δένδρο που έχει βάθος d και έχει παράγοντα διακλάδωσης b :



Βάθος: d

Στην λίστα ΑΝΟΙΚΤΕΣ θα αποθηκευτούν το πολύ $b-1$ κόμβοι σε κάθε κόμβο του μονοπατιού. Άρα η πολ/τα είναι:

$$O(bd)$$

Η πολυπλοκότητα είναι πολυωνυμική

Β. Θεωρία

2. Αλγόριθμοι Τυφλής Αναζήτησης

2. Αλγόριθμος Αναζήτησης Κατά Πλάτος (1.Ψευδογλώσσα)

- Ο αλγόριθμος διάσχισης κατά πλάτος κινείται κατά επίπεδα στον χώρο αναζήτησης προσπαθώντας να εντοπίσει μία λύση:

ΑΛΓΟΡΙΘΜΟΣ ΚΑΤΑ ΠΛΑΤΟΣ

Βήμα 1: Δημιούργησε το δένδρο αναζήτησης T , το οποίο αρχικά αποτελείται από τον κόμβο αρχής n_0

Βήμα 1.1: Δημιούργησε μια **ουρά** και ονόμασε την **ΑΝΟΙΚΤΕΣ**

Βήμα 1.2: Δημιούργησε μια **λίστα** και ονόμασε την **ΚΛΕΙΣΤΕΣ**

Βήμα 1.3: Τοποθέτησε τον n_0 στην ΑΝΟΙΚΤΕΣ

Βήμα 2: Αν η ΑΝΟΙΚΤΕΣ είναι κενή ουρά τότε τερμάτισε με αποτυχία

Βήμα 2.1: Επέλεξε το πρώτο στοιχείο της και αφαίρεσε το από αυτήν

Βήμα 2.2: Ονόμασε αυτό το στοιχείο n

Βήμα 3: Αν το n είναι ο στόχος τότε τερμάτισε με επιτυχία και επέστρεψε τη λύση.

Βήμα 3.1: **Επέκτεινε** το n , δηλαδή εμφάνισε τα παιδιά της που δεν είναι στις κλειστές

Βήμα 3.2: Τοποθέτησε τους παραγόμενους κόμβους στο τέλος της ουράς.

Βήμα 3.3: Βάλε το n στις ΚΛΕΙΣΤΕΣ και αφαίρεσε όλες τις εμφανίσεις του n από τις ΑΝΟΙΚΤΕΣ

Βήμα 4: Επέστρεψε στο βήμα 2.

Β. Θεωρία

2. Αλγόριθμοι Τυφλής Αναζήτησης

2. Αλγόριθμος Αναζήτησης Κατά Πλάτος (1.Ψευδογλώσσα)

Εμπειρικά:

ΑΛΓΟΡΙΘΜΟΣ ΚΑΤΑ ΠΛΑΤΟΣ

Αρχικά:

- Βάζουμε την αφετηρία στο δένδρο.

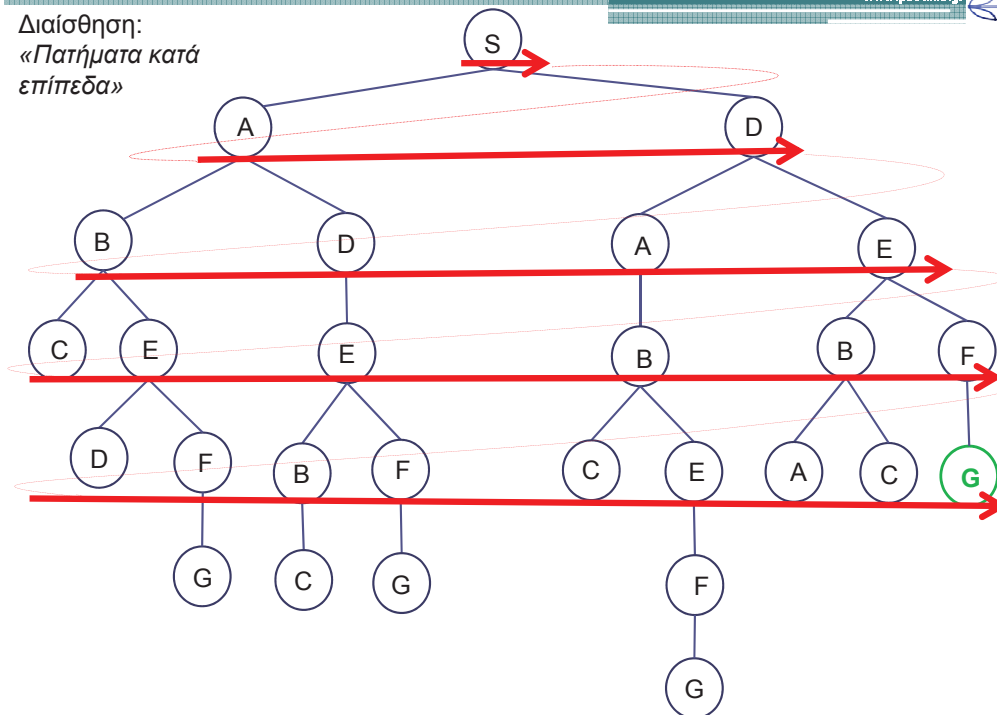
Επαναληπτικά:

- **Πατάμε** «κατά επίπεδα» στον επόμενο κόμβο.
- **Ανοίγουμε** τους γείτονες του (που δεν είναι πρόγονοί του) στον γράφο και τους θέτουμε ως παιδιά του.
- **Διαγράφουμε** ανοικτές εμφανίσεις του κόμβου που πατήσαμε.

Εως ότου:

- Πατήσουμε στον κόμβο-στόχο

Διαίσθηση:
«Πατήματα κατά
επίπεδα»



Β. Θεωρία

2. Αλγόριθμοι Τυφλής Αναζήτησης

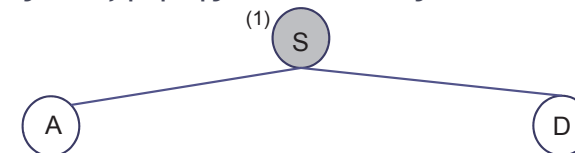
2. Αλγόριθμος Αναζήτησης Κατά Πλάτος (2.Παράδειγμα Εκτέλεσης)



Β. Θεωρία

2. Αλγόριθμοι Τυφλής Αναζήτησης

2. Αλγόριθμος Αναζήτησης Κατά Πλάτος (2.Παράδειγμα Εκτέλεσης)



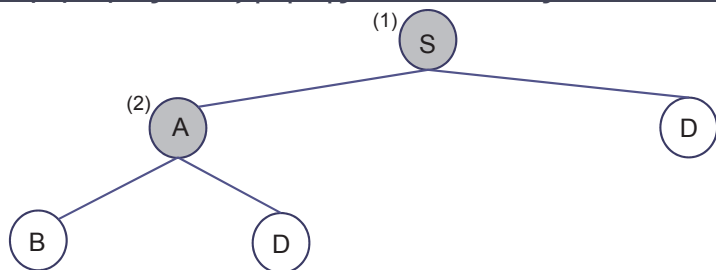
Βήμα	ΑΝΟΙΚΤΕΣ	ΚΛΕΙΣΤΕΣ
0		(S){}

Βήμα	ΑΝΟΙΚΤΕΣ	ΚΛΕΙΣΤΕΣ
0		(S){}
1		(A,D){S}

Β. Θεωρία

2. Αλγόριθμοι Τυφλής Αναζήτησης

2. Αλγόριθμος Αναζήτησης Κατά Πλάτος (2.Παράδειγμα Εκτέλεσης)

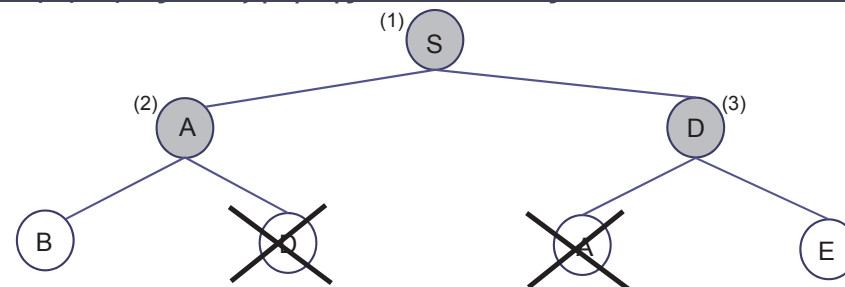


Βήμα	ΑΝΟΙΚΤΕΣ	ΚΛΕΙΣΤΕΣ
0	(S)	{}
1	(A,D)	{S}
2	(D _S , B, D _{SA})	{S,A}

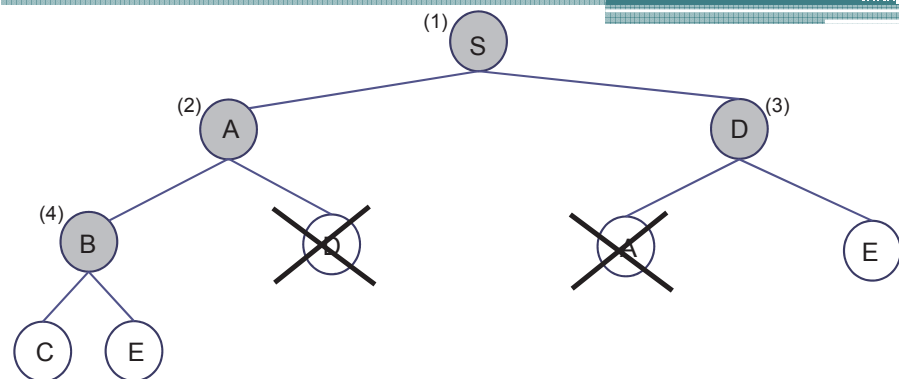
Β. Θεωρία

2. Αλγόριθμοι Τυφλής Αναζήτησης

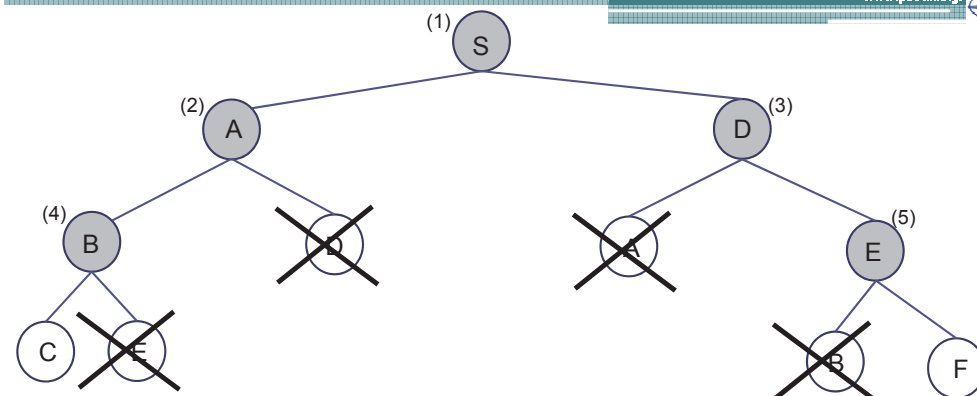
2. Αλγόριθμος Αναζήτησης Κατά Πλάτος (2.Παράδειγμα Εκτέλεσης)



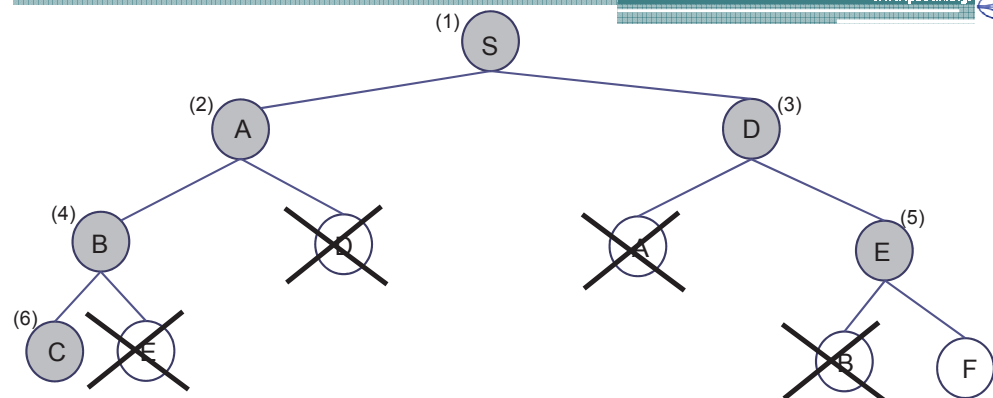
Βήμα	ΑΝΟΙΚΤΕΣ	ΚΛΕΙΣΤΕΣ
0	(S)	{}
1	(A,D)	{S}
2	(D _S , B, D _{SA})	{S,A}
3	(B,E)	{S,A,D}



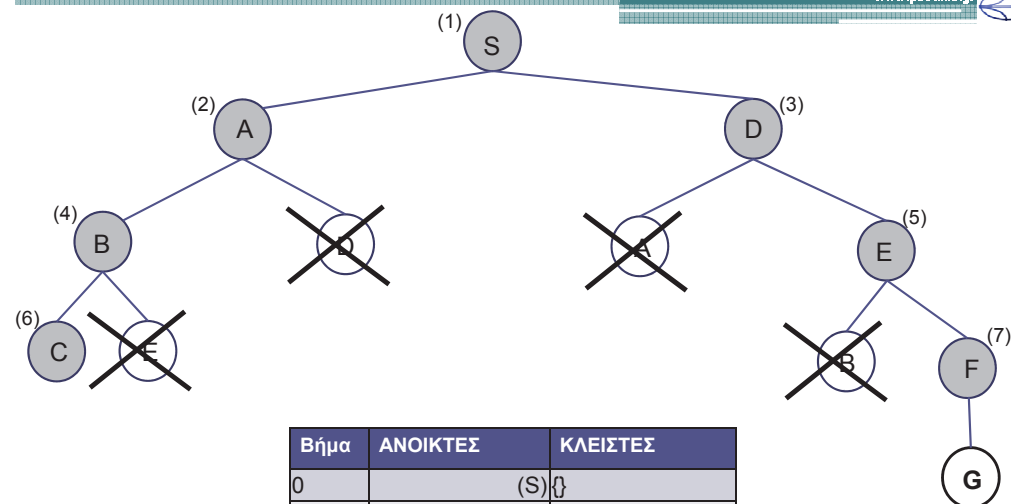
Βήμα	ΑΝΟΙΚΤΕΣ	ΚΛΕΙΣΤΕΣ
0	(S)	{}
1	(A,D)	{S}
2	(D _S , B, D _{SA})	{S,A}
3	(B,E)	{S,A,D}
4	(E _{SD} , C, E _{SAB})	{S,A,D,B}



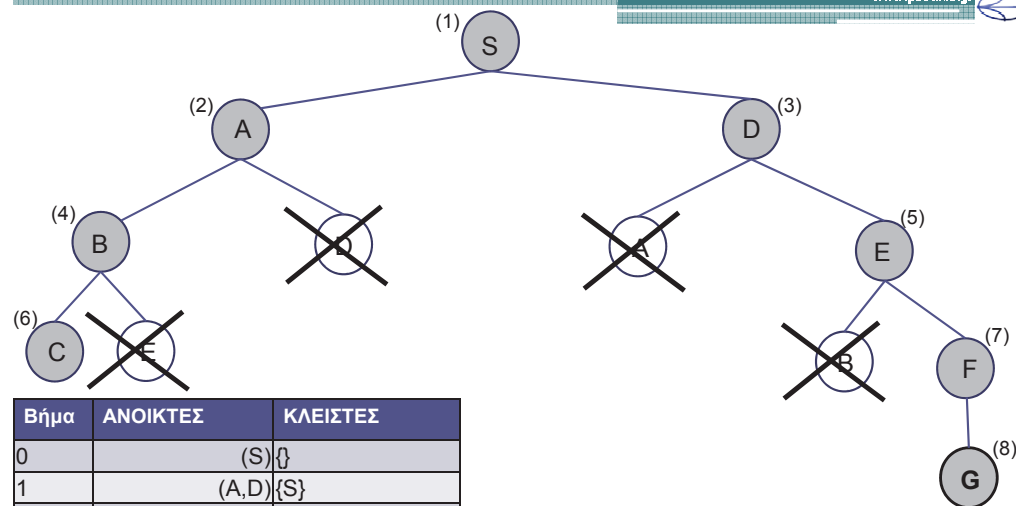
Βήμα	ΑΝΟΙΚΤΕΣ	ΚΛΕΙΣΤΕΣ
0	(S)	{}
1	(A,D)	{S}
2	(D _S , B, D _{SA})	{S,A}
3	(B,E)	{S,A,D}
4	(E _{SD} , C, E _{SAB})	{S,A,D,B}
5	(C,F)	{S,A,D,B,E}



Βήμα	ΑΝΟΙΚΤΕΣ	ΚΛΕΙΣΤΕΣ
0	(S)	{}
1	(A,D)	{S}
2	(D _S , B, D _{SA})	{S,A}
3	(B,E)	{S,A,D}
4	(E _{SD} , C, E _{SAB})	{S,A,D,B}
5	(C,F)	{S,A,D,B,E}
6	(F)	{S,A,D,B,E,C}



Βήμα	ΑΝΟΙΚΤΕΣ	ΚΛΕΙΣΤΕΣ
0	(S)	{}
1	(A,D)	{S}
2	(D _S , B, D _{SA})	{S,A}
3	(B,E)	{S,A,D}
4	(E _{SD} , C, E _{SAB})	{S,A,D,B}
5	(C,F)	{S,A,D,B,E}
6	(F)	{S,A,D,B,E,C}
7	(G)	{S,A,D,B,E,C,F}



Βήμα	ΑΝΟΙΚΤΕΣ	ΚΛΕΙΣΤΕΣ
0	(S)	{}
1	(A,D)	{S}
2	(D _S , B, D _{SA})	{S,A}
3	(B,E)	{S,A,D}
4	(E _{SD} , C, E _{SAB})	{S,A,D,B}
5	(C,F)	{S,A,D,B,E}
6	(F)	{S,A,D,B,E,C}
7	(G)	{S,A,D,B,E,C,F}
8	()	{S,A,D,B,E,C,F,G}

Μονοπάτι: S-D-E-F-G
Κόστος Μονοπατιού: 13

Σειρά Επίσκεψης: S-A-D-B-E-C-F-G
Βήματα: 8

Β. Θεωρία

2. Αλγόριθμοι Τυφλής Αναζήτησης

1. Αλγόριθμος Αναζήτησης Κατά Πλάτος (3. Παρατηρήσεις)

ΠΑΡΑΤΗΡΗΣΗ 1: Σειρά Τοποθέτησης των Παιδιών

- Υιοθετούμε την εξής σύμβαση: Όταν εισάγουμε τα παιδιά ενός κόμβου η σειρά με την οποία τα βάζουμε στην λίστα ΑΝΟΙΚΤΕΣ είναι λεξικογραφική (αλφαβητική)
- Αν ακολουθήσουμε διαφορετική σύμβαση θα προκύψει διαφορετική εκτέλεση της κατά βάθος που ίσως οδηγήσει σε άλλη λύση
 - Αν δεν καθορίζεται από την εκφώνηση ακολουθούμε την λεξικογραφική σειρά και θα αναφέρουμε ότι κάνουμε την συγκεκριμένη παραδοχή.
 - Σε κάποιες εκφωνήσεις θα μας δίνεται η σειρά των κόμβων για την διάσπαση των ισοπαλιών, δηλαδή για την απόφαση με ποια σειρά εισάγονται οι κόμβοι στην λίστα ΑΝΟΙΚΤΕΣ

B. Θεωρία

2. Αλγόριθμοι Τυφλής Αναζήτησης

1. Αλγόριθμος Αναζήτησης Κατά Πλάτος (3. Παρατηρήσεις)

ΠΑΡΑΤΗΡΗΣΗ 2: Παραλλαγές του Αλγορίθμου Κατά Πλάτος

- Το ΕΑΠ δεν έχει διαλευκάνει το κριτήριο τερματισμού του αλγορίθμου
- Συγκεκριμένα ακολουθούνται δύο κριτήρια τερματισμού:
 - Να πατήσουμε στον κόμβο-στόχο (όπως στις διαφάνειες αυτές)
 - Οπότε και σταματάμε στο βήμα 8
- Να εμφανιστεί ο κόμβος-στοχος στους απογόνους του κόμβου που «πατάμε»
 - Στο παράδειγμα μας θα σταματούσαμε στο βήμα 7.

B. Θεωρία

2. Αλγόριθμοι Τυφλής Αναζήτησης

2. Αλγόριθμος Αναζήτησης Κατά Πλάτος (4. Πληρότητα)

ΠΛΗΡΟΤΗΤΑ ΑΛΓΟΡΙΘΜΟΥ ΑΝΑΖΗΤΗΣΗΣ

Ένας αλγόριθμος αναζήτησης θα λέμε ότι είναι πλήρης, εάν σίγουρα μπορεί να εντοπίσει μια (οποιαδήποτε) λύση στον χώρο αναζήτησης εφόσον αυτή υπάρχει.
Εάν δεν υπάρχει θα πρέπει να είναι σε θέση να απαντήσει ότι δεν υπάρχει λύση.

- Ο αλγόριθμος αναζήτησης κατά πλάτος ΕΙΝΑΙ πλήρης, διότι έστω και σε έναν πολύ μεγάλο χρόνο θα εξετάσει όλους τους κόμβους του δένδρου.

B. Θεωρία

2. Αλγόριθμοι Τυφλής Αναζήτησης

2. Αλγόριθμος Αναζήτησης Κατά Πλάτος (5. Βέλτιστος)

Πότε ένας ΑΛΓΟΡΙΘΜΟΣ ΑΝΑΖΗΤΗΣΗΣ είναι ΒΕΛΤΙΣΤΟΣ

Ένας αλγόριθμος αναζήτησης θα λέμε ότι είναι ΒΕΛΤΙΣΤΟΣ με βάση δύο κριτήρια:

A: Αν επιστρέφει την λύση που είναι κοντινότερα στην ρίζα (Αν ο Γράφος δεν έχει Βάρη).

B: Αν επιστρέφει την λύση που έχει το μικρότερο κόστος (Αν ο Γράφος έχει Βάρη).

- Ο αλγόριθμος αναζήτησης Κατά Πλάτος:
 - Όσον αφορά την απόσταση από την ρίζα: Είναι βέλτιστος διότι θα εντοπίσει την βέλτιστη λύση σε απόσταση από την ρίζα.
 - Όσον αφορά το κόστος: Δεν είναι βέλτιστος, διότι δεν λαμβάνει καν υπόψιν τα κόστη.

B. Θεωρία

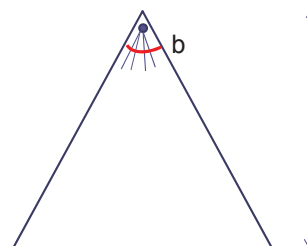
2. Αλγόριθμοι Τυφλής Αναζήτησης

2. Αλγόριθμος Αναζήτησης Κατά Πλάτος (6. Πολ/τα Χρόνου)

ΧΡΟΝΙΚΗ ΠΟΛΥΠΛΟΚΟΤΗΤΑ ενός ΑΛΓΟΡΙΘΜΟΥ ΑΝΑΖΗΤΗΣΗΣ

Χρονική πολυπλοκότητα ενός αλγορίθμου αναζήτησης είναι το πλήθος των κόμβων του χώρου αναζήτησης που απαιτείται να εξετάσει ο αλγόριθμος προκειμένου να οδηγηθεί σε μία λύση.

- Κάνουμε μια ανάλυση χειρότερης περίπτωσης σε ένα δένδρο που έχει βάθος d και έχει παράγοντα διακλάδωσης b:



Βάθος: d

Στην χειρότερη περίπτωση ο αλγόριθμος θα εξετάσει όλους τους κόμβους του δένδρου:

$$1 + b + b^2 + \dots + b^d = \frac{b^{d+1} - 1}{b - 1} = O(b^{d+1})$$

Η πολυπλοκότητα είναι εκθετική

Β. Θεωρία

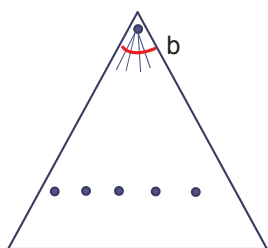
2. Αλγόριθμοι Τυφλής Αναζήτησης

2. Αλγόριθμος Αναζήτησης Κατά Πλάτος (7. Πολ/τα Χώρου)

ΧΩΡΙΚΗ ΠΟΛΥΠΛΟΚΟΤΗΤΑ ενός ΑΛΓΟΡΙΘΜΟΥ ΑΝΑΖΗΤΗΣΗΣ

Χωρική πολυπλοκότητα ενός αλγόριθμου αναζήτησης είναι το πλήθος των κόμβων του χώρου αναζήτησης που θα πρέπει να αποθηκεύονται στην λίστα ΑΝΟΙΚΤΕΣ.

- Κάνουμε μια ανάλυση χειρότερης περίπτωσης σε ένα δένδρο που έχει βάθος d και έχει παράγοντα διακλάδωσης b :



Βάθος: d

Στην λίστα ΑΝΟΙΚΤΕΣ θα αποθηκευτούν το πολύ τόσοι κόμβοι όσο το πλήθος των κόμβων του τελευταίου επιπέδου. Άρα η πολ/τα είναι:

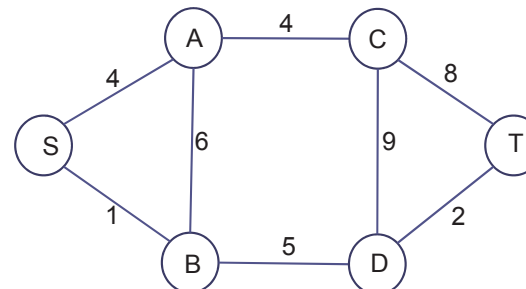
$$O(b^d)$$

Η πολυπλοκότητα είναι εκθετική

Γ. Ασκήσεις

Ασκηση Κατανόησης 1

(α) Σχεδιάστε τον χώρο αναζήτησης του παρακάτω γραφήματος θεωρώντας ως αφετηρία τον κόμβο S



(β) Πόσα τα μονοπάτια από το S στο T

(γ) Ποιο το βέλτιστο μονοπάτι από το S στο T

Γ. Ασκήσεις

Ασκηση Κατανόησης 2

Εκτελέστε τον αλγόριθμο αναζήτησης κατά βάθος στον χώρο αναζήτησης της άσκησης κατανόησης 1 και κατασκευάστε το δένδρο αναζήτησης που προκύπτει, θεωρώντας ως κόμβο στόχο τον T

Γ. Ασκήσεις

Ασκηση Κατανόησης 3

Εκτελέστε τον αλγόριθμο αναζήτησης κατά πλάτος στον χώρο αναζήτησης της άσκησης κατανόησης 1 και κατασκευάστε το δένδρο αναζήτησης που προκύπτει, θεωρώντας ως κόμβο στόχο τον T



Ε. Ασκήσεις

Εφαρμογή 1

Στον γράφο-καταστάσεων της εφαρμογής 1 του ΜΑΘ1.1 εκτελέστε τον αλγόριθμο κατά βάθος, έτσι ώστε να βρεθεί μια διαδρομή που συνδέει τους δύο παρακάτω κόμβους:



Προτεραιότητα έχει η μετακίνηση του κύβου Α, μετά του Β και μετά του Γ. Για τον ίδιο μετακινούμενο κύβο, προτεραιότητα έχει η μετακίνηση του πάνω σε άλλους κύβους, με αλφαβητική προτεραιότητα και έπειτα η μετακίνησή του προς το τραπέζι (εφόσον δεν είναι ήδη εκεί).



Ε. Ασκήσεις

Εφαρμογή 2

Στο πρόβλημα της πλοήγησης του Robbie στον Λαβύρινθο της εφαρμογής 3 του μαθήματος 1.1:

(Α) Εκτελέστε τον αλγόριθμο κατά βάθος για την εύρεση ενός μονοπατιού από την αφετηρία (δωμάτιο S) στο δωμάτιο F

(Β) Εκτελέστε τον αλγόριθμο κατά πλάτος για την εύρεση ενός μονοπατιού από την αφετηρία (δωμάτιο S) στο δωμάτιο F



Ε. Ασκήσεις

Εφαρμογή 3

Η εταιρία «Νευρωνικά Ρομπότ» παρουσιάζει το νέο της μοντέλο, τον πλοηγό πάρκων NP-310. Το NP-310 είναι δημοφιλές γιατί όπου και αν είσαι μέσα στο πάρκο σου λέει πώς πρέπει να κινηθείς για να φτάσεις όπου θες μέσα στο πάρκο χωρίς να πατήσεις το πράσινο. Οι κινήσεις που μπορεί να υποδείξει το NP-310 στο χρήστη του είναι βόρεια (↑), νότια (↓), ανατολικά (→), ή δυτικά (←).

Στο παρακάτω σχήμα φαίνεται το σχήμα του ενός πάρκου, όπου οι σκιασμένες περιοχές αντιστοιχούν σε πράσινο και τα υπόλοιπα είναι δρόμοι.

		Y				
		1	2	3	4	5
X	1					
	2					
	3					
	4					
	5					

Σημειώνουμε πως λόγω προβλημάτων στην ακρίβεια του μηχανισμού προσδιορισμού θέσης, το NP-310 μπορεί να προσδιορίσει τη θέση του με ακρίβεια 50 μέτρων αλλά το παραπάνω σχήμα του πάρκου (το οποίο έχει μέγεθος 250m x 250m) είναι ακριβές για το σκοπό που το θέλουμε. Η εταιρία «Νευρωνικά Ρομπότ», παράλληλα με την παρουσίαση του NP-310 προκήρυξε και ένα διαγωνισμό για την ανακάλυψη του μηχανισμού πλοήγησης που χρησιμοποιεί το NP-310 προκειμένου να φτάσει από ένα τετράγωνο-αφετηρία σε ένα τετράγωνο-προορισμό.



(Α) Σχεδιάστε τον γράφο καταστάσεων του προβλήματος



(Β) Ποιά είναι η διαδρομή από το τετράγωνο $(,1)$ στο τετράγωνο $(4,1)$ στην αναζήτηση-κατά-βάθος; Υποθέστε ότι μεταξύ εναλλακτικών επιλέγετε πρώτα αυτή με τη μικρότερη X συντεταγμένη (και μετά, αν χρειαστεί, αυτή με τη μικρότερη Y συντεταγμένη).



(Γ) Ποιά είναι η διαδρομή από το τετράγωνο $(3,3)$ στο τετράγωνο $(4,1)$ στην αναζήτηση-κατά-πλάτος; Υποθέστε ότι μεταξύ εναλλακτικών επιλέγετε πρώτα αυτή με τη μικρότερη X συντεταγμένη (και μετά, αν χρειαστεί, αυτή με τη μικρότερη Y συντεταγμένη).