

Ασκήσεις μελέτης της 3^{ης} διάλεξης

3.1. Επιλέξτε τις σωστές απαντήσεις, μόνο μία σε κάθε ερώτηση.

α) Ο αλγόριθμος αναζήτησης **πρώτα σε βάθος** έχει:

καλύτερη χειρότερη την ίδια
πολυπλοκότητα **χρόνου** συγκρινόμενος με τον αλγόριθμο αναζήτησης **πρώτα σε πλάτος**.

Η πολυπλοκότητα χρόνου του DFS είναι $O(b^m)$, ενώ του BFS είναι $O(b^{d+1})$. Στη χειρότερη περίπτωση, ο DFS ψάχνει όλο το χώρο αναζήτησης, μέχρι το μέγιστο βάθος m , που μπορεί να είναι και άπειρο, ενώ ο BFS ψάχνει μέχρι ένα επίπεδο πιο κάτω από το βάθος d της ρηχότερης τελικής κατάστασης.

β) Ο αλγόριθμος αναζήτησης **πρώτα σε βάθος χωρίς κλειστό σύνολο** έχει:

καλύτερη χειρότερη την ίδια
πολυπλοκότητα **χώρου** συγκρινόμενος με τον αλγόριθμο αναζήτησης **πρώτα σε πλάτος χωρίς κλειστό σύνολο**.

Η πολυπλοκότητα χώρου του DFS χωρίς κλειστό σύνολο είναι $O(bm)$, δηλαδή γραμμική, ενώ του BFS χωρίς κλειστό σύνολο είναι $O(b^{d+1})$, δηλαδή εκθετική ως προς το βάθος της ρηχότερης τελικής κατάστασης d . Θα μπορούσε, βέβαια, κανείς να παρατηρήσει ότι για άπειρο m , ο DFS ενδέχεται να μην τερματίσει.

γ) Αν **υπάρχει** λύση και ο μέγιστος παράγοντας διακλάδωσης είναι πεπερασμένος και το σύνολο καταστάσεων είναι πεπερασμένο και το κόστος λύσης είναι αύξουσα συνάρτηση του βάθους (και μόνο), ο αλγόριθμος αναζήτησης **πρώτα σε βάθος με κλειστό σύνολο**:

βρίσκει πάντα λύση και μάλιστα βέλτιστη δεν βρίσκει πάντα λύση
 βρίσκει πάντα λύση, αλλά όχι σίγουρα βέλτιστη.

Αφού το σύνολο των (δυνατών) καταστάσεων είναι πεπερασμένο, τα μόνα άπειρα κλαδιά του δέντρου αναζήτησης που είναι δυνατόν να προκύψουν (και στα οποία θα μπορούσε να παγιδευτεί ο DFS) αντιστοιχούν σε κύκλους του γράφου καταστάσεων. Αφού, όμως, χρησιμοποιείται κλειστό σύνολο, ο DFS δεν θα παγιδευτεί ποτέ σε τέτοια άπειρα κλαδιά (κύκλους του γράφου). Επομένως, αφού και ο μέγιστος παράγοντας διακλάδωσης είναι πεπερασμένος, το δέντρο αναζήτησης που θα ψάξει ο DFS είναι πεπερασμένο (όλα τα άπειρα κλαδιά priονίζονται, αφαιρώντας τις επαναλήψεις κύκλων του γράφου) και ο DFS θα τερματίσει σίγουρα, είτε όταν συναντήσει τελική κατάσταση είτε όταν ψάξει ανεπιτυχώς ολόκληρο το δέντρο. Αφού όμως υπάρχει σίγουρα λύση, υπάρχει σίγουρα και τουλάχιστον μία τελική κατάσταση (το τέλος του μονοπατιού της λύσης). Επομένως, ο DFS θα τερματίσει συναντώντας τελική κατάσταση, οπότε και θα αναφέρει τη λύση (μονοπάτι) ως εκείνη την τελική κατάσταση. Η λύση, όμως, που θα αναφέρει ενδέχεται να μην είναι βέλτιστη, γιατί ενδέχεται να υπάρχει και άλλη τελική κατάσταση σε μικρότερο βάθος.

δ) Αν **δεν** υπάρχει λύση και ο μέγιστος παράγοντας διακλάδωσης είναι πεπερασμένος και το σύνολο καταστάσεων είναι πεπερασμένο, ο αλγόριθμος αναζήτησης **πρώτα σε βάθος με κλειστό σύνολο**:

τερματίζει πάντα δεν τερματίζει ποτέ
 άλλοτε τερματίζει και άλλοτε δεν τερματίζει.

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.

ε) Ο αλγόριθμος αναζήτησης **επαναληπτικής εκβάθυνσης χωρίς κλειστό σύνολο** έχει:
X καλύτερη ___ χειρότερη ___ την ίδια
πολυπλοκότητα **χώρου** συγκρινόμενος με τον αλγόριθμο αναζήτησης **πρώτα σε πλάτος χωρίς κλειστό σύνολο**.

Χωρίς κλειστό σύνολο, ο IDS έχει πολυπλοκότητα χώρου $O(bd)$, ενώ ο BFS έχει $O(b^{d+1})$. Επομένως, ο IDS έχει καλύτερη πολυπλοκότητα χώρου.

στ) Ο αλγόριθμος αναζήτησης **επαναληπτικής εκβάθυνσης χωρίς κλειστό σύνολο** έχει:
X καλύτερη ___ χειρότερη ___ την ίδια
πολυπλοκότητα **χώρου** συγκρινόμενος με τον αλγόριθμο αναζήτησης **πρώτα σε βάθος χωρίς κλειστό σύνολο**.

Χωρίς κλειστό σύνολο, ο IDS έχει πολυπλοκότητα χώρου $O(bd)$, ενώ ο DFS έχει $O(bm)$. Επομένως, αφού $d \leq m$, ο IDS έχει εν γένει καλύτερη πολυπλοκότητα χώρου.