Εισαγωγη στον Σχεδιασμο και Αναλυση Αλγοριθμων
Ανοιξη 2010
2ο Πακετο Ασκησεων
Ενδεικτικες Λυσεις

1. **Ιδέα:** Ένα γράφημα είναι διμερές ανν είναι 2-χρωματίσιμο. Ένας απλός τρόπος λοιπόν να δούμε αν ένα γράφημα είναι διμερές είναι να το χρωματίσουμε με 2 χρώματα, και μετά να δούμε αν υπάρχει έστω και μια ακμή που έχει το ίδιο χρώμα στα δύο τις άκρα. Αν δεν υπάρχει σημαίνει πως το γράφημα είναι 2-χρωματίσιμο, και άρα διμερές.

   Ο χρωματισμός όμως δεν πρέπει να είναι εντελώς τυχαίος αλλιώς το αποτέλεσμα δεν θα είναι έγκυρο.

   Μια πολύ απλή παρατήρηση είναι πως αν το γράφημα είναι 2-χρωματίσιμο, τότε η παρακάτω διαδικασία παράγει ένα έγκυρο χρωματισμό:

   1) Ξεκίνα από μια τυχαία κορυφή, έστω s, και δώσε της ένα χρώμα, έστω κόκκινο.

   2) Κάθε κορυφή που απέχει περιττή απόσταση από την s χρωμάτισέ την με το άλλο χρώμα και κάθε κορυφή που απέχει άρτια απόσταση από το s χρωμάτισέ την κόκκινη.

   Είναι εύκολο να δούμε πως ο παραπάνω κανόνας είναι σωστός, και αυτόν ακριβώς θα εκμεταλεύτουμε. Θα τρέξουμε BFS με μια μικρή προσθήκη του χρωματισμού.

   Για κάθε $v \in V(G)$
           χρώμα(v) $= NULL$
           $c(v)$=άσπρο
           $d(v) = \infty$

   $c(s)$ =γκρι
   $d(s) = 0$
   χρώμα(s)=κόκκινο
   Βάλε(Q, s)

   Όσο $Q \neq \emptyset$
       k=Βγάλε(Q)
       Για κάθε $v \in Adj(k)$
           Αν $c(v)$=άσπρο

$c(v)$=γκρι
$d(v) = d(k) + 1$

Άν $d(v)$ είναι άρτιος
χρώμα(v)=κόκκινο
αλλιώς
χρώμα(v)=μπλε

Βάλε(Q, v) (Τέλος "Για κάθε")
$c(k)$ =μαύρο (Τέλος "Όσο")

Για κάθε $(u, v) \in E(G)$
Άν χρώμα(u)=χρώμα(v)
Επέστρεψε("Όχι")
Επέστρεψε("Ναι")

Ο χρόνος εκτέλεσης του αλγορίθμου είναι ο χρόνος της BFS συν τον χρόνο που χρειάζεται ο έλεγχος όλων των ακμών δηλαδή $O((V + E) + E) = O(V + E)$

2. (α′) $T(n) = 2T(n/2) + n$. Από Κ.Θ $T(n) = O(nlogn)$

   (β′) Η αναδρομή σταματά όταν το $n = 1$. Αφού σε κάθε βήμα η είσοδος υποδιπλασιάζεται, έχουμε ότι ο αλγόριθμος σταματά όταν $n/2^k = 1$ άρα $k = logn$

   (γ′) Το δένδρο έχει n φύλλα!

   (δ′) Ο αλγόριθμος της συγχωνευτικής ταξινόμησης (μεταξύ άλλων) λειτουργεί ακριβώς όπως ο πιο πάνω αλγόριθμος.

3. Τρέξτε τον αλγόριθμο BFS σε μια τυχαία κορυφή του γραφήματος. Αν το γράφημα είναι συνεκτικό η αναζήτηση αυτή θα φτάσει σε όλες τις κορυφές, δηλαδή ο πίνακας d που αποθηκεύει τις αποστάσεις από το s, δεν έχει κανένα στοιχείο $= \infty$. Οπότε τρέξτε τον BFS και απλά κάντε τον επίπλεον έλεγχο αυτό, ο χρόνος παραμένει γραμμικός.

4. Ο $O(n^2)$ είναι άνω φράγμα στον χρόνο εκτέλεσης ενός αλγορίθμου. Άρα "Ο χρόνος εκτέλεσης του τάδε αλγορίθμου είναι το πολύ $O(n^2)$"

5. Ο αλγόριθμος που ψάχνεται είναι η δυαδική αναζήτηση. Περιγραφικά:
   1. Κοίτα το μεσαίο στοιχείο
   2. Αν είναι ίσο με το k, τέλος
   3. Αν είναι μεγαλύτερο από το k, τρέξε ξανά τον αλγόριθμο με τον αριστερό μισό πίνακα.
   4. Αν είναι μικρότερο από το k, τρέξε ξανά τον αλγόριθμο με τον δεξιό μισό πίνακα.

   Η αναδρομική σχέση που εκφράζει τον χρόνο εκτέλεσης του αλγορίθμου είναι:
   $T(n) = T(n/2) + O(1)$, δηλαδή $O(logn)$