3η Άσκηση

Ημερομηνία Παράδοσης: 3 Ιουνίου 2012

Πρόβλημα 1 Για την άσκηση θα πρέπει να συμπληρώσετε τον κώδικα μίας συνάρτησης η οποία, δεδομένων δύο κόμβων σε ένα δυαδικό δέντρο αναζήτησης (Binary Search Tree), υπολογίζει το μονοπάτι από τον πρώτο κόμβο στον δεύτερο. Το μονοπάτι αυτό επιστρέφεται ως διπλά συνδεδεμένη λίστα (doubly connected linked list). Η συνάρτησή σας έχει δήλωση:

```
List* find_BST_path(BST_node* root, BST_node* start, BST_node* end);
```

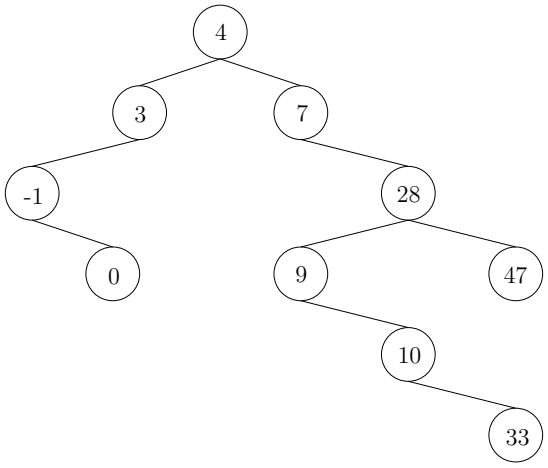
Για διευκόλυνσή σας σας δίνονται επτά αρχεία

- `BST.h`: header αρχείο με τις δηλώσεις για το δυαδικό δέντρο αναζήτησης. **Αυτό το αρχείο δεν πρέπει να το αλλάξετε**.
- `BST.c`: c αρχείο με τις υλοποιήσεις για το δυαδικό δέντρο αναζήτησης. **Αυτό το αρχείο δεν πρέπει να το αλλάξετε**.
- `list.h`: header αρχείο με τις δηλώσεις για τη διπλά συνδεδεμένη λίστα. **Αυτό το αρχείο δεν πρέπει να το αλλάξετε**.
- `list.c`: c αρχείο με τις υλοποιήσεις για τη διπλά συνδεδεμένη λίστα. **Αυτό το αρχείο δεν πρέπει να το αλλάξετε**.
- `find_BST_path.h`: header αρχείο με τη δήλωση για τη συνάρτηση που πρέπει να γράψετε. **Αυτό το αρχείο δεν πρέπει να το αλλάξετε**.
- `find_BST_path.c`: c αρχείο όπου πρέπει να συμπληρώσετε την δική σας υλοποίηση. **Αυτό είναι το αρχείο πρέπει να το αλλάξετε και να προσθέσετε τη δική σας υλοποίηση**.
- `find_path.c`: c αρχείο με τη συνάρτηση `main` που περιέχει παράδειγμα με το οποίο μπορείτε να ελέγξετε τον κώσικά σας. **Αυτό είναι το αρχείο μπορείτε να το αλλάξετε ώστε να ελέγξετε το πρόγραμμά σας**.

Η συνάρτησή σας θα πρέπει να επιστρέφει `NULL` αν έχετε το οποιοδήποτε πρόβλημα στη δυναμική διαχείριση μνήμης, είτε αν οι κόμβοι `start` και `end` δεν ανήκουν στο δέντρο με ρίζα `root`. Επίσης οι δείκτες `start`, `end` και `root` δεν πρέπει να είναι `NULL`.

Για παράδειγμα αν σας δοθεί η ακολουθία αριθμών $\{4, 7, 3, 28, 9, 10, 47, 33, -1, 0\}$, το δυαδικό δέντρο που προκύπτει αν τους προσθέσετε σε αυτό από αριστερά προς τα δεξιά είναι το δέντρο που φαίνεται στην εικόνα 1. Αν τώρα ζητήσετε το μονοπάτι από τον κόμβο με κλειδί 10 στον κόμβο με κλειδί 47, η συνάρτηση `find_BST_path` θα πρέπει να επιστρέψει λίστα με τους κόμβους με κλειδιά $10, 9, 28$ και 47. Αντίστοιχα, αν ζητήσετε το μονοπάτι από τον κόμβο με κλειδί 47 στον κόμβο με κλειδί 10, η συνάρτηση `find_BST_path` θα πρέπει να επιστρέψει λίστα με τους κόμβους με κλειδιά $47, 28, 9$ και 10. Αν πάλι ζητήσετε το μονοπάτι από τον κόμβο με κλειδί 10 στον κόμβο με κλειδί 7, η συνάρτηση `find_BST_path` θα πρέπει να επιστρέψει λίστα με τους κόμβους με κλειδιά $10, 9, 28$ και 7. Για να βρείτε το μονοπάτι που σας ζητείται προτείνεται ο παρακάτω αλγόριθμος. Έστω `start` και `end` ο αρχικός και τελικός κόμβος του μονοπατιού που ψάχνετε. Τότε ακολουθείτε τα παρακάτω βήματα

1. Βρίσκουμε τον κόμβο v με το μικρότερο ύψος που είναι πρόγονος τόσο του `start` όσο και του `end`. Το κόμβος αυτός χαρακτηρίζεται από την ιδιότητα ότι: είτε (1) είναι ο `start`, είτε (2) είναι ο `end`, είτε (3) οι κόμβοι `start` και `end` είναι σε διαφορετικά υποδέντρα αυτού (δηλαδή, ο `start` είναι στο αριστερό υποδέντρο του v, ενώ ο `end` στο δεξί υποδέντρο του v, ή το ανάποδο), ενώ δεν υπάρχει απόγονός του με την ίδια ιδιότητα. Για παράδειγμα, για τους κόμβους με κλειδιά 10 και 47, ο κόμβος v είναι ο κόμβος με κλειδί 28, ενώ για τους κόμβους με κλειδιά 10 και 7, είναι ο κόμβος με κλειδί 7.



Σχήμα 1: Το δυαδικό δέντρο αναζήτησης για τα δεδομένα του παραδείγματος.

2. Αν ο v είναι ο κόμβος `start`, τότε το μονοπάτι που ψάχνουμε προκύπτει αν ξεκινήσουμε από τον `end`, και ανεβαίνουμε προς τα πάνω μέχρι να βρούμε τον `start`, και στη συνέχεια αντιστρέψουμε το μονοπάτι αυτό.
3. Αν ο v είναι ο κόμβος `end`, τότε το μονοπάτι που ψάχνουμε προκύπτει αν ξεκινήσουμε από τον `start`, και ανεβαίνουμε προς τα πάνω μέχρι να βρούμε τον `end`.
4. Τέλος, αν ο κόμβος v δεν είναι ούτε ο κόμβος `start`, ούτε ο κόμβος `end`, το μονοπάτι που ψάχνουμε είναι το μονοπάτι από τον `start` στον v, και από τον `end` στο v, αντεστραμμένο όμως. για παράδειγμα για τους κόμβους 10 και 47, όπου ο v είναι ο 28, το πρώτο μονοπάτι είναι το $10 \to 9 \to 28$, και το δεύτερο το $47 \to 28$. το δεύτερο πρέπει να το αντιστρέψουμε (ώστε να γίνει $28 \to 47$, και μετά να συνδέσουμε τα δύο μονοπάτια (προσέχοντας φυσικά να μην εμφανιστεί ο κόμβος 28 δύο φορές).

Για να παραδώσετε την άσκηση αρκεί να στείλετε email στον διδάσκοντα με το αρχείο `find_BST_path.c` συνημμένο (και μόνο το αρχείο αυτό). Το μήνυμά σας θα πρέπει να έχει subject 'em102 hwk3¨, και θα πρέπει στο body να γράψετε το ονοματεπώνυμό σας, τον αριθμό μητρώου σας και το login σας στα μηχανήματα του τμήματος (μην στείλετε το password σας). Η παράδοση πρέπει να γίνει μέχρι τις 23:59 στις 3 Ιουνίου 2012.