Lab 5: Binary Search Trees

EPL231 – Data Structures and Algorithms

Αναδρομή

- Η αναδρομή εμφανίζεται όταν μία διεργασία καλεί τον εαυτό της
- Αποτελείται από δυο κομμάτια:
  - Μία τερματική κατάσταση π.χ.
    - `if (i==0)`
  - Μία αναδρομική κατάσταση όπου η διεργασία καλεί τον εαυτό της με διαφορετικούς παραμέτρους

Παράδειγμα αναδρομής

- Η συνάρτηση int power(int x, int pow)
- Επιτρέπεται μόνο η χρήση του πολλαπλασιασμού
- Επιστρέφει το x^{pow}
- Η σκέψη
  - $x^0=1$
  - $x^2=x \cdot x$ $(x^1 \cdot x^1)$
  - $x^5=x \cdot x \cdot x \cdot x \cdot x$ $(x^4 \cdot x)$

Παράδειγμα αναδρομής (συνέχεια)

- Ποια είναι η τερματική κατάσταση;
  - `pow==0` όπου η συνάρτηση επιστρέφει 1
  - Άρα κάπου στον κώδικα θα υπάρχει
  - `if(pow==0) return 1;`
- Ποια είναι η αναδρομική κατάσταση;
  - `pow!=0`, τι γίνεται σε αυτή την περίπτωσή;
  - Πρέπει με κάποιο τρόπο να φτάσουμε στο `pow=0`.
  - Αυτό γίνεται π.χ. αν αφαιρούμε 1 από το pow κάθε φορά

Παράδειγμα αναδρομής (συνέχεια)

- Άρα κάπου θα έχουμε κάλεσμα της συνάρτησης με pow-1
- Δεδομένο: κάθε φορά θέλουμε να πολλαπλασιάζουμε το x.
- Βάζοντας μαζί ότι σκεφτήκαμε:

```
int power(int x, int pow){
     if(pow==0)
          return 1;
     else
          return x * power(x, pow-1)
}
```

Δυαδικό Δέντρο Αναζήτησης

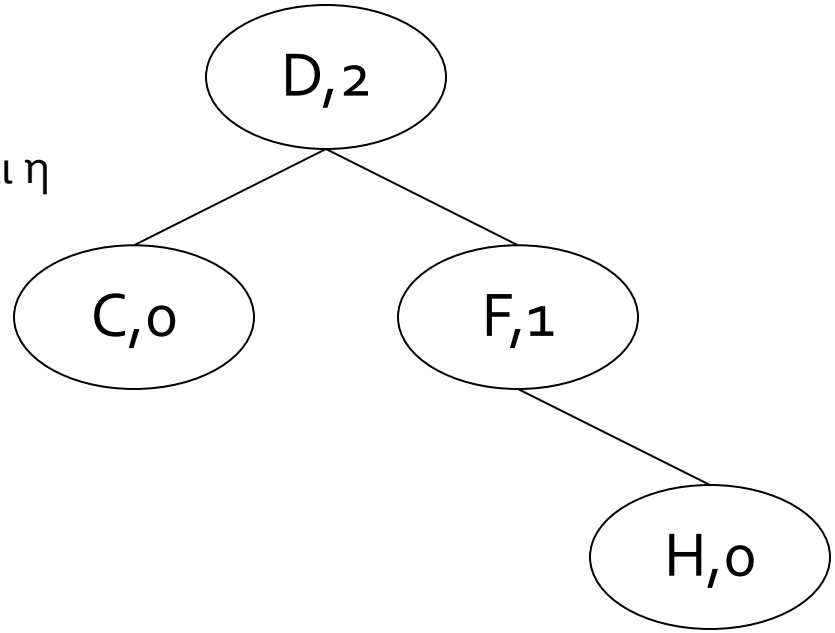
- Δυαδικό Δέντρο:
  - Μια δομή όπου ο κάθε κόμβος κρατάει κάποια δεδομένα καθώς επίσης και δέικτες προς δυο άλλα δυαδικά δέντρα (αριστερά και δεξιά υποδέντρα)
- Δυαδικό Δέντρο Αναζήτησης (ΔΔΑ):
  - Ένα δυαδικό δέντρο, όπου οι τιμές του ενός υποδέντρου περιέχουν τιμές μικρότερες ή ίσες από την τιμή του πατέρα τους και το άλλο υποδέντρο κρατάει τις υπόλοιπες τιμές.

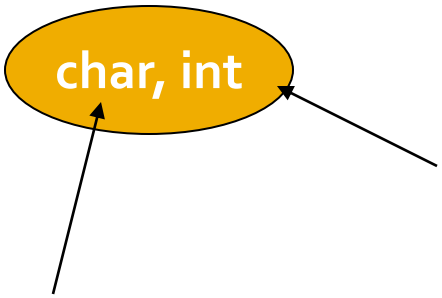
Δυαδικό Δέντρο Αναζήτησης με Αλφαβητική Σειρά

Μετά από εισαγωγή του D, C, F, H

Τι διαφέρει αν αρχίζαμε από την αρχή και η σειρά εισαγωγής ήταν:

1. D, F, C, H
2. D, C, H, F
3. C, D, F, H



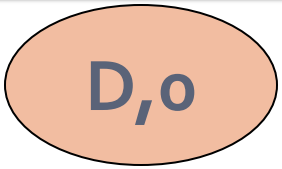


Ακέραιος αριθμός που αντιπροσωπεύει το ύψος του κόμβο.

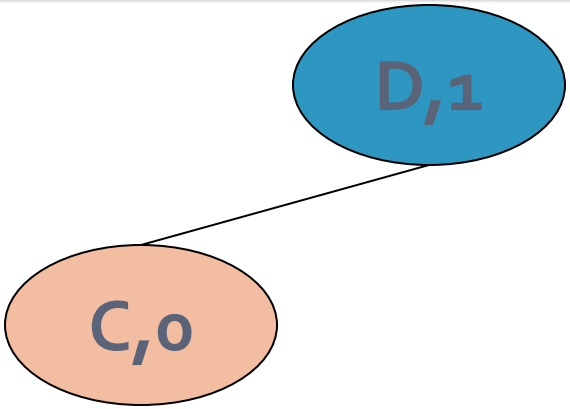
Ένας χαρακτήρας του αγγλικού αλφαβήτου

Παράδειγμα Εκτέλεσης

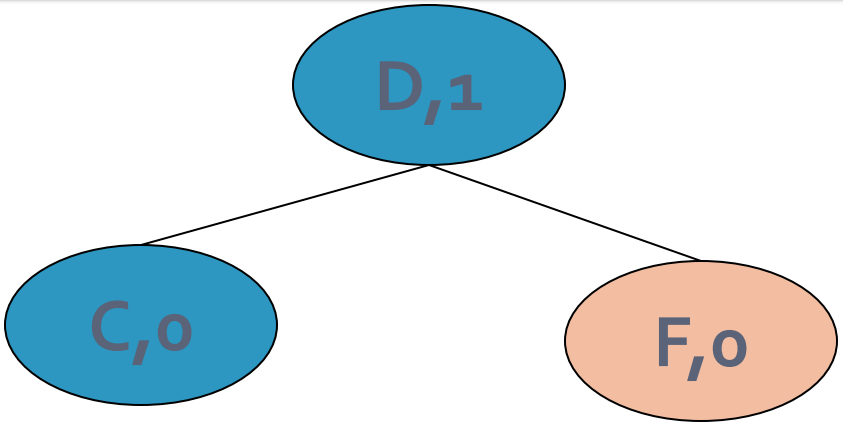
Εισαγωγή του κόμβου D



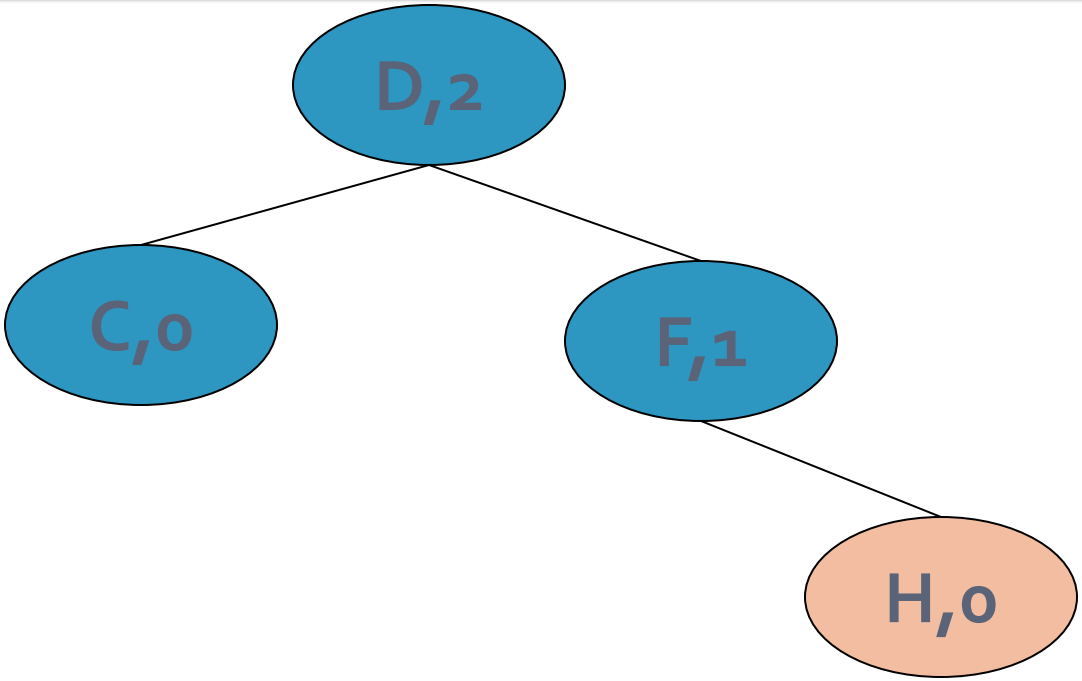
Εισαγωγή του κόμβου C



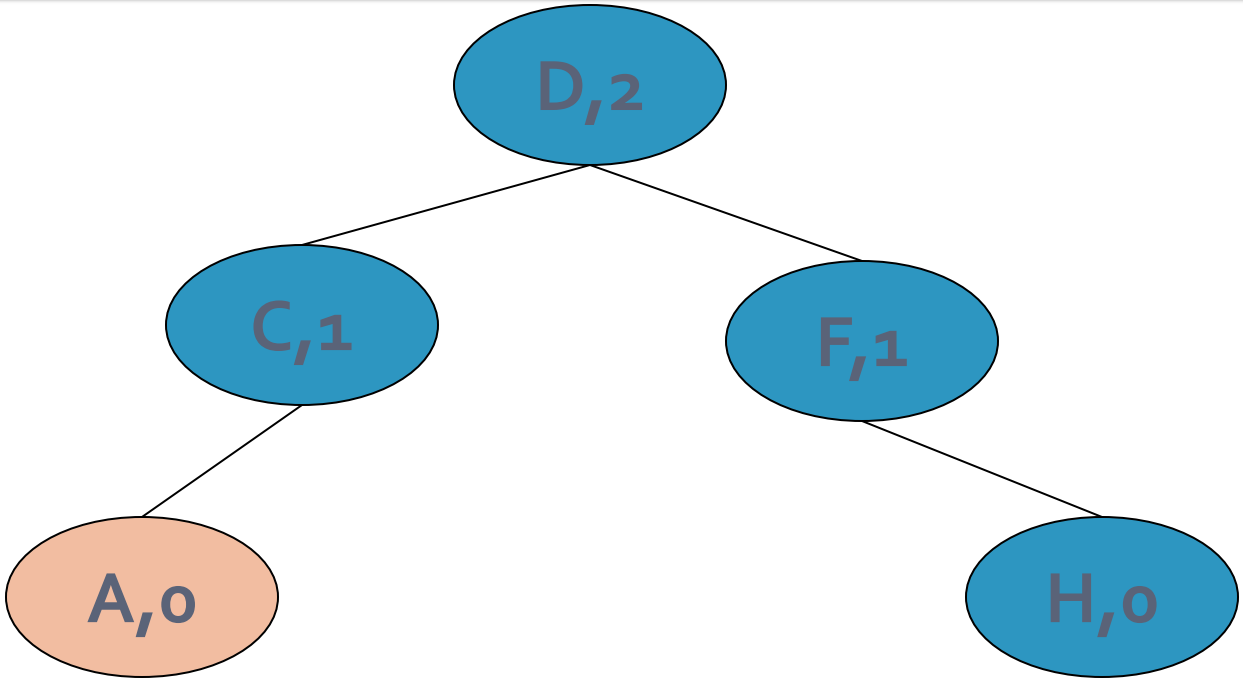
Εισαγωγή του κόμβου F



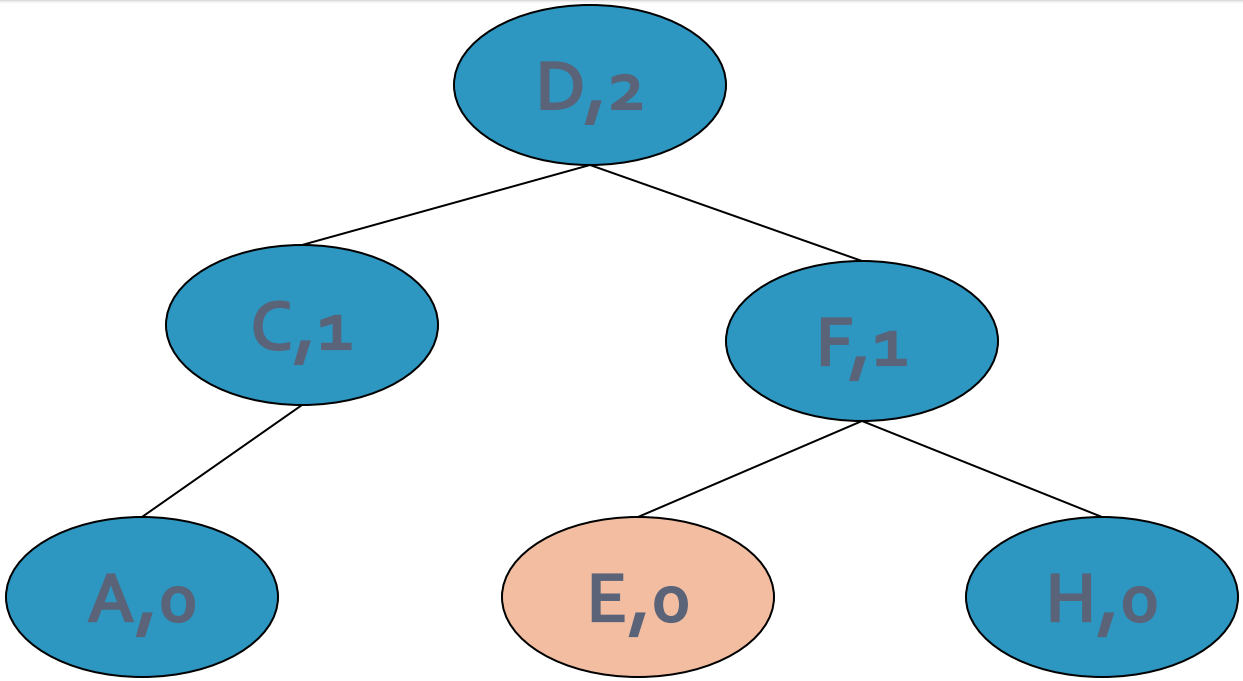
Εισαγωγή του κόμβου H



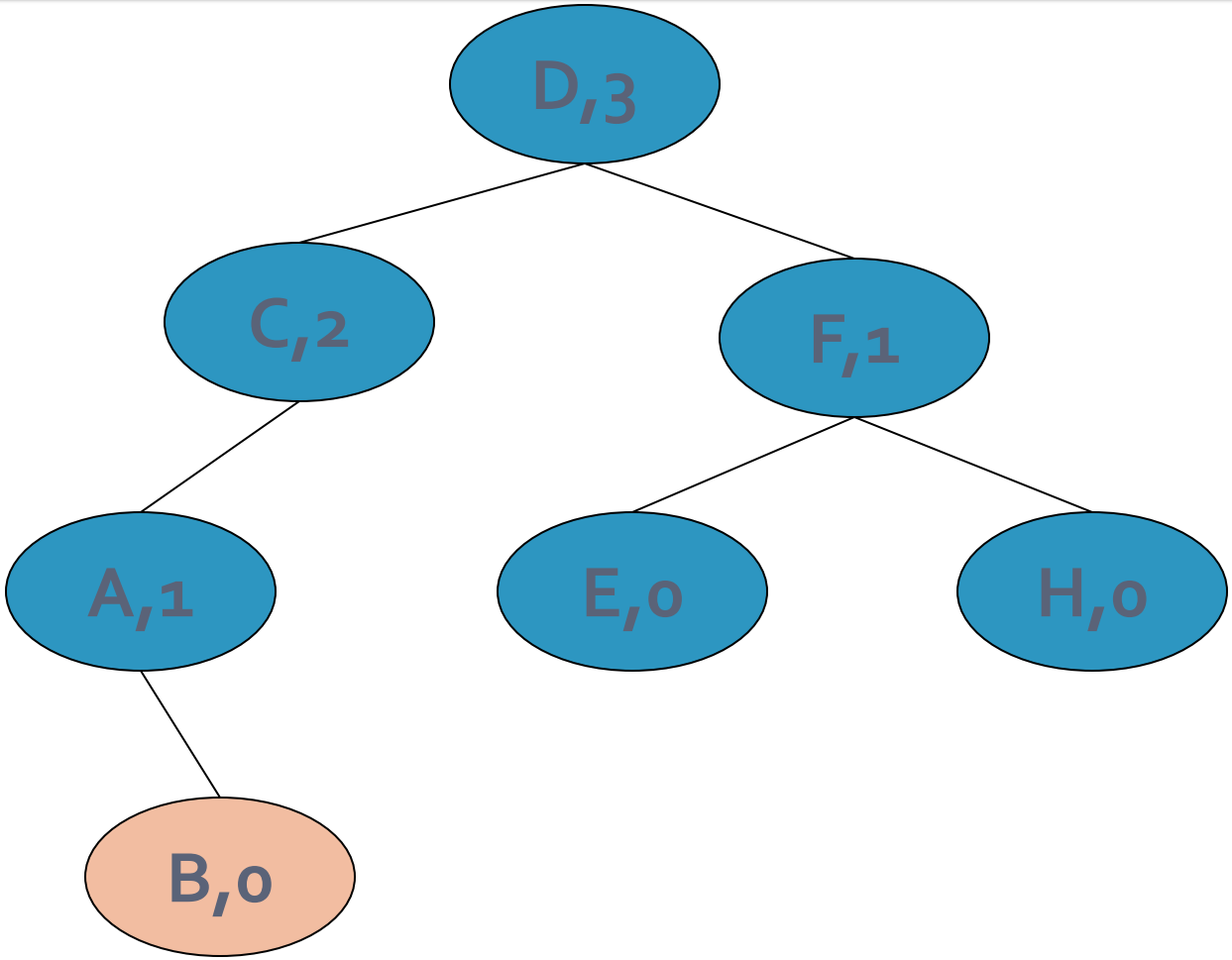
Εισαγωγή του κόμβου A



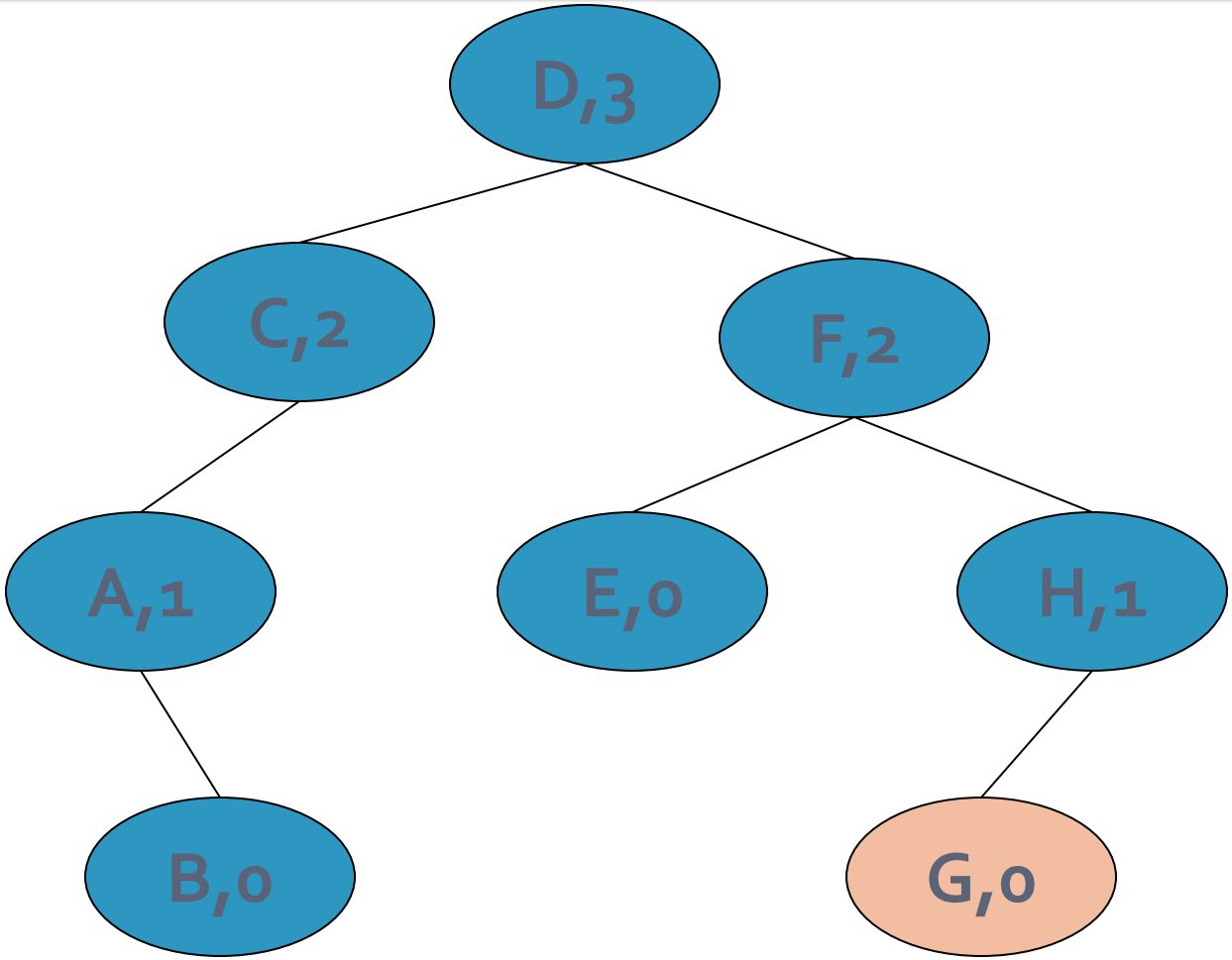
Εισαγωγή του κόμβου Ε



Εισαγωγή του κόμβου Β



Εισαγωγή του κόμβου G



Δομές (Structures)

```
typedef struct _BSTreeNode {
  Type data;
  int height;
  struct _BSTreeNode *leftChild;
  struct _BSTreeNode *rightChild;
  struct _BSTreeNode *father;
} BSTreeNode;


typedef struct _BSTree {
  BSTreeNode *root;
} BSTree;


typedef BSTreeNode *BSTreeNodePtr;
typedef BSTree *BSTreePtr;
```

Συναρτήσεις προς υλοποίηση

- Εισαγωγή Δεδομένων:
  - `void` **`insertData`**`(BSTree *tree, Type value)`
  - `void` **`insertNode`**`(BSTreeNode *root, BSTreeNode *node)`
    - Χρησιμοποιείστε την συνάρτηση `getSubtreeHeight()` για να ανανεώσετε το ύψος των κόμβων
- Εύρεση Δεδομένων:
  - `BSTreeNode *`**`findData`**`(BSTree *tree, Type value)`
  - `BSTreeNode *`**`findNode`**`(BSTreeNode *root, Type value)`

The End