

Δομές Δεδομένων

Κωνσταντίνος Τσερπές

(βασισμένο στις διαφάνειες του κου Δημήτριου Μιχαήλ)

Δομές Δεδομένων

- Μια δομή δεδομένων είναι μια συλλογή δεδομένων με κάποιες ιδιότητες η οποία προσφέρει εύκολη πρόσβαση σε αυτά τα δεδομένα.
- Έχουμε δει ως τώρα μια στατική δομή δεδομένων, **τον πίνακα**.
- Τι κάνουμε όμως στην περίπτωση που θέλουμε να αποθηκεύουμε ένα συνεχώς μεταβαλλόμενο αριθμό από δεδομένα;

Δυναμικές Δομές Δεδομένων

- Είναι δομές δεδομένων που μπορούν να αυξομειώνουν το μέγεθος τους, δηλαδή να αναπτύσσονται και να συρρικνώνονται στον χρόνο εκτέλεσης.
- Η C ως γλώσσα προγραμματισμού δεν παρέχει τέτοιες δομές. Μας παρέχει όμως τα απαραίτητα εργαλεία για να φτιάξουμε τέτοιες δομές δεδομένων

Απαραίτητα Εργαλεία

- structs - εγγραφές
 - δυναμική καταχώρηση μνήμης - malloc, free
 - δείκτες - pointers
 - μετονομασία τύπων – typedef
-
- Για να φτιάξουμε δυναμικές δομές δεδομένων χρειάζεται να χρησιμοποιήσουμε τα πιο σημαντικά και συχνά πιο δυσκολονόητα κομμάτια της γλώσσας.

Διάφορες Δομές Δεδομένων

- Θα δούμε δύο από τις πιο σημαντικές δομές δεδομένων
 - συνδεδεμένες λίστες
 - στοίβες
- Ανάλογα με τις ανάγκες της εφαρμογής χρησιμοποιούμε διαφορετική δομή.

Αυτοαναφερόμενες Εγγραφές

- Η C μας παρέχει την δυνατότητα να φτιάξουμε εγγραφές (structs) οι οποίες αυτοαναφέρονται:

```
struct node {  
    int data;  
    struct node *next;  
};
```

- Η δομή node περιέχει έναν ακέραιο και ένα δείκτη τύπου node.

Αυτοαναφερόμενες Δομές

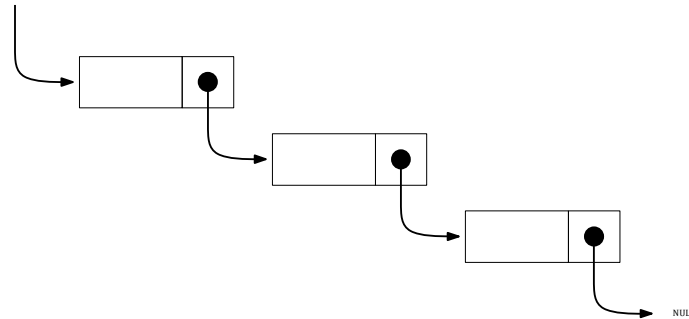
- Ουσιαστικά η C μας επιτρέπει να δηλώσουμε μια μεταβλητή δείκτη σε τύπο που δεν έχει οριστεί ακόμη. Για παράδειγμα το παρακάτω πρόγραμμα είναι απόλυτα σωστό.

```
#include <stdio.h>
int main ( ) {
    struct node *ptr ;
}
```

- Προσέξτε πως στο παραπάνω πρόγραμμα δεν δηλώνουμε ποτέ την εγγραφή node.

Συνδεδεμένη Λίστα

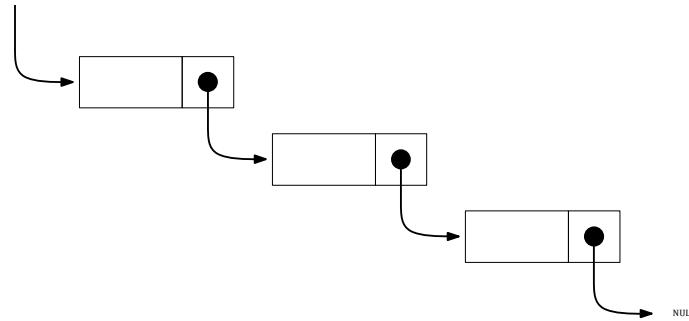
- Μια γραμμική συλλογή αυτοαναφερόμενων εγγραφών, που ονομάζονται κόμβοι, και συνδέονται από συνδέσεις δεικτών.



- Μια συνδεδεμένη λίστα προσπελάζεται μέσω ενός δείκτη στον πρώτο κόμβο της λίστας. Κατά σύμβαση, ο δείκτης σύνδεσης στον τελευταίο κόμβο της λίστας τίθεται σε NULL για να σηματοδοτήσει το τέλος της λίστας.

Συνδεδεμένη λίστα

- Η πληροφορία σε κάθε κόμβο μπορεί να είναι οποιαδήποτε τύπου, ακόμη και struct.



Συνδεδεμένη λίστα: Κόμβος

```
typedef struct {  
    int AM ;  
    char name [50] ;  
}student;  
  
struct node {  
    student data;  
    struct node *next;  
};
```

- Ένας κόμβος λίστας που περιέχει πληροφορίες για έναν φοιτητή.

Υλοποίηση λίστας

[lists.c](#)