

# Διάλεξη 22η: Διπλά Συνδεδεμένες Λίστες

Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης

Εισαγωγή στην Επιστήμη Υπολογιστών



# Διπλά συνδεδεμένη λίστα

- Οργάνωση των δεδομένων
  - Θεωρούμε τα δεδομένα διατεταγμένα γραμμικά (το ένα μετά το άλλο)
  - Κάθε στοιχείο δείχνει στο επόμενο στη μνήμη ή στο **NULL** αν είναι το τελευταίο
  - Κάθε στοιχείο δείχνει και στο προηγούμενο στη μνήμη ή στο **NULL** αν είναι το πρώτο
- Παράδειγμα δήλωσης (λίστα ακεραίων)

dllist.c

```
node_t {  
    int data;  
    node_t *next;  
    node_t *prev;  
};  
typedef struct list_node node_t;  
node_t *root;
```



## • Τελεστές

- Δημιουργία κενής λίστας
- Εισαγωγή στοιχείου πριν από άλλο στοιχείο
- Εισαγωγή στοιχείου μετά από άλλο στοιχείο
- Εύρεση συγκεκριμένου στοιχείου
- Διαγραφή στοιχείου
- Διάσχιση της λίστας
- Απελευθέρωση/διαγραφή λίστας



## ● Αλγόριθμοι

- Δημιουργία κενής λίστας  
Ο κενός δείκτης `NULL` αναπαριστά την κενή λίστα
- Εισαγωγή στοιχείου πριν/μετά από άλλο στοιχείο  
`node_t *list_insert_before(node_t *list, int data);`  
`void list_insert_after(node_t *position, int data);`
- Εύρεση συγκεκριμένου στοιχείου  
`node_t *list_find(node_t *list, int data);`  
`node_t *list_nth(node_t *list, int position);`
- Διαγραφή στοιχείου  
`node_t *list_remove_node(node_t *node);`  
`node_t *list_remove_data(node_t *list, int data);`
- Διάσχιση της λίστας  
`void list_traverse(node_t *list, void (*f)(node_t *node));`  
`void list_print(node_t *list);`
- Απελευθέρωση/διαγραφή λίστας  
`void list_delete(node_t *list);`



# Εισαγωγή στοιχείου (1)

- Εισαγωγή στοιχείου πριν από άλλο στοιχείο

dllist.c

```
node_t *list_insert_before(node_t *list, int data)
{
    node_t *node;

    node = (node_t *) malloc(sizeof(node_t));
    node->data = data;
    node->next = list;
    if (list == NULL) {
        node->prev = NULL;
    } else {
        node->prev = list->prev;
        list->prev = node;
    }
    if (node->prev != NULL) {
        node->prev->next = node;
    }
    return node;
}
```



# Εισαγωγή στοιχείου (2)

- Εισαγωγή στοιχείου μετά από άλλο στοιχείο

dllist.c

```
void list_insert_after(node_t *position, int data)
{
    node_t *node;

    if(position == NULL) {
        return list_insert_before(NULL, data);
    }
    node = list_insert_before(position->next, data);
    position->next = node;
    node->prev = position;
}
```



- Διαγραφή στοιχείου

dllist.c

```
node_t *list_remove_node(node_t *node)
{
    node_t *list;

    if(node->next != NULL) {
        node->next->prev = node->prev;
        list = node->next;
    }
    if(node->prev != NULL) {
        node->prev->next = node->next;
        list = node->prev;
    }
    return list;
}
```

