



# Αλγόριθμοι και Δομές Δεδομένων(Θ)

## Ενότητα 10: ΣΥΝΔΕΔΕΜΕΝΕΣ ΛΙΣΤΕΣ

Ευάγγελος Γ. Ούτσιος

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΕ



# Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο ΤΕΙ Κεντρικής Μακεδονίας» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



# ΣΥΝΔΕΔΕΜΕΝΕΣ ΛΙΣΤΕΣ

## *Στοίβα ως Συνδεδεμένη Λίστα*

Η στοίβα υλοποιείται με έναν δείκτη `top` που αρχικοποιείται στο `NULL`, και δείχνει ότι η στοίβα είναι άδεια.

Η λειτουργία `push` δεν ελέγχει για υπερχείλιση γιατί θεωρητικά η στοίβα μπορεί να έχει όσους κόμβους θέλουμε (αφού δημιουργείται δυναμικά).

Επίσης, η λειτουργία `pop`, με τη βοήθεια της εντολής `free` επιστρέφει στη διαθέσιμη μνήμη του Η/Υ το χώρο που καταλαμβάνονταν από τον κόμβο που διαγράφηκε.

# ΣΥΝΔΕΔΕΜΕΝΕΣ ΛΙΣΤΕΣ

```
PTR top = NULL;
```

```
PTR push(int obj, PTR t)
```

```
{  
    PTR newnode;  
    newnode = malloc(sizeof(struct  
        node));  
    newnode->data = obj;  
    newnode->next = t;  
    t = newnode;  
    return t;  
}
```

```
PTR pop(PTR t, int *obj)
```

```
{  
    PTR p;  
    if (t == NULL)  
    {  
        printf("Stack empty.\n");  
        getch();  
    }  
    else  
    {  
        p = t;  
        t = t->next;  
        *obj = p->data;  
        free(p);  
    }  
    return t;  
}
```

# ΣΥΝΔΕΔΕΜΕΝΕΣ ΛΙΣΤΕΣ

## *Ουρά ως Συνδεδεμένη Λίστα*

Η ουρά υλοποιείται με τη χρήση δύο δεικτών front και rear που αρχικοποιούνται στην τιμή NULL.

Η άδεια ουρά εκφράζεται με  $front = NULL$ . Όπως και στην περίπτωση της στοίβας, δεν χρειάζεται να γίνεται έλεγχος υπερχείλισης από τη στιγμή που η ουρά αυξάνεται δυναμικά.

# ΣΥΝΔΕΔΕΜΕΝΕΣ ΛΙΣΤΕΣ

```
PTR front = NULL, rear = NULL;
void enqueue(int obj, PTR *pf, PTR *pr)
{
    PTR newnode;
    newnode = malloc(sizeof(struct
        node));
    newnode->data = obj;
    newnode->next = NULL;
    if ((*pf) == NULL)
    {
        *pf = newnode;
        *pr = newnode;
    }
    else
    {
        (*pr)->next = newnode;
        *pr = newnode;
    }
}
```

```
void dequeue(PTR *pf, PTR *pr)
{
    PTR p;
    if ((*pf) == NULL)
        printf("\nQueue empty. No
            elements to delete.\n");
    else
    {
        p = *pf;
        *pf = (*pf)->next;
        if ((*pf) == NULL)
            *pr = *pf;
        printf("\n%d has been
            deleted...\n", p->data);
        free(p);
    }
    getch();
}
```