# Δυναμικός προγραμματισμός

Βασικά χαρακτηριστικά:

- Διάσπαση του αρχικού προβλήματος σε υποπροβλήματα
- Για την επίλυση ενός υποπροβλήματος χρησιμοποιείται η λύση των προηγούμενων υποπροβλημάτων.
- Οι λύσεις των προηγούμενων υποπροβλημάτων αποθηκεύονται στην μνήμη π.χ. σε κάποιο πίνακα ή μεταβλητή.
- Συσχέτιση των υποπροβλημάτων, με κόμβους γραφήματος και των εξαρτήσεών τους, με κατευθυνόμενες ακμές.
- Παρατήρηση: Η συσχέτιση ενός προβλήματος Δ.Π με γράφημα, δεν πρέπει να συγχέεται με την επίλυση. Παρά μόνο με την σειρά επίλυσης των υποπροβλημάτων (τοπολογική ταξινόμηση).
- Πολυπλοκότητα αλγορίθμων Δ.Π. εξαρτάται από το # υποπροβλημάτων και των ΣΧΒ που εκτελούνται σε καθένα.

Παράδειγμα: Να βρεθεί ο n-οστός όρος της ακολουθίας Fibonacci.

Έστω αναδρομική συνάρτηση $F(n)$.

$$F(n) = \begin{cases} n>1: & F(n-1)+F(n-2), \\ n=1: & 1 \\ n=0: & 0 \end{cases}$$

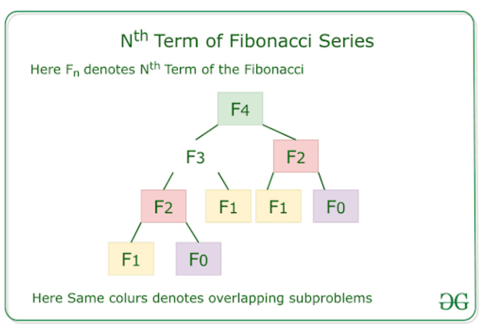
π.χ. $F(3) = F(2)+F(1) = F(1)+F(0)+F(1) = 1+0+1$

Καλείται δύο φορές η συνάρτηση $F()$ για $n=1$

Άρα δεν αποτελεί επίλυση Δ.Π.

```
// Fibonacci Series using Recursion

int fib(int n)
{
    if (n <= 1)
        return n;
    return fib(n - 1) + fib(n - 2);
}
```




Up to Bottom

## Επίλυση με Δ.Π.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| i: | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| f | 0 | 1 | 1 | 2 | 3 | 5 | 8 |

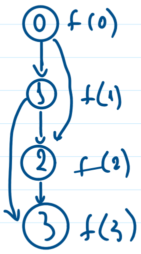
$$f[n] = f[n-1] + f[n-2]$$

Για τον υπολογισμό του 6ου όρου της ακολουθίας χρησιμοποιούμε...

Τα υποπροβλήματα $f(i)$ ως κόμβοι ενός γραφήματος είναι τοπολογικά ταξινομημένοι σε αύξουσα σειρά, εφόσον οι κόμβοι ορίζονται με τον δείκτη $i$

$$f(0) \rightarrow f(1) \rightarrow f(2) \rightarrow f(3)$$

Bottom to up.




```c
int fib(int n)
{
    /* Declare an array to store Fibonacci numbers. */
    int f[n + 2]; // 1 extra to handle case, n = 0
    int i;

    f[0] = 0;
    f[1] = 1;

    for (i = 2; i <= n; i++) {
        f[i] = f[i - 1] + f[i - 2];
    }

    return f[n];
}
```

$\Rightarrow$ $f[i] \rightarrow$ $i$-οστό υποπρόβλημα.

## SOSARA!!!

Για την επίλυση ενός προβλήματος Δ.Π. ακολουθούμε πάντα τα παρακάτω βήματα:

1) Ορισμός Υποπροβλημάτων: Ως υποπρόβλημα $f(i), i \in \{0, \dots, n\}$ ορίζω το πρόβλημα εύρεσης του $i$-οστού όρου της ακολουθίας Fibonacci.

2) Καθορισμός επιλογών: Μία μόνο επιλογή έχω για την επίλυση κάθε επιμέρους υποπροβλ. $f(i)$. Η λύση του $f(i)$ να προκύψει από την πρόσθεση των λύσεων για $f(i-1)$ και $f(i-2)$

3) Ορισμός Αναδρομικής σχέσης: $f(i) = \begin{cases} f(i-1) + f(i-2), & i = \{2, \dots, n\} \\ 1, & i = 1 \\ 0, & i = 0 \end{cases}$

4) Τοπολογική ταξινόμηση: Τα υποπροβλήματα $f(i)$ ως κόμβοι ενός γραφήματος είναι τοπολογικά ταξινομημένοι σε αύξουσα σειρά, εφόσον οι κόμβοι ορίζονται με τον δείκτη $i$

5) Επίλυση Αρχικού προβλήματος: Η λύση του προβλήματος για $i = n$, βρίσκεται αποθηκευμένη στο $f[n]$