

Recursion

Si suddivide un problema in sottoproblemi **simili ma più semplici**.

Esempio: calcolo del numero triangolare n-esimo (https://en.wikipedia.org/wiki/Triangular_number)

per $n=4$, $Nt(n) = 10$

```
[]  
[][]  
[][][]  
[][][][]
```

Funzione che calcola il numero triangolare n-esimo.

```
int numeroTriangolare(int n) {  
    int result;  
  
    .....  
  
    return result;  
}
```

Se $n=1$, $Nt(n) = 1$ **CASO BASE e CLAUSOLA DI CHIUSURA**

Per n generico, $Nt(n) = n \cdot Nt(n - 1)$

```
int numeroTriangolare(int n) {  
    int result;  
    if (n == 1)  
        result = 1;    // caso base  
    else  
        result = n + numeroTriangolare(n - 1); // ricorsione  
  
    return result;  
}
```

REQUISITI

- ogni invocazione ricorsiva deve semplificare l'elaborazione
- devono esistere casi speciali che 'chiudano' la ricorsione.

nota: occorre gestire le situazioni anomale (es $n = -1$)

Tipologie di ricorsione

- Ricorsione diretta. Esempio: fattoriale
- Ricorsione multipla. Esempio: fibonacci (https://en.wikipedia.org/wiki/Fibonacci_number)
- Ricorsione indiretta. Esempio: $f(n) = k_1 * g(n-1); g(n) = k_2 * f(n-1)$

Problemi con la Ricorsione

- spazio sprecato
- complessità alta
- ricorsione infinita (*dovuta ad errori di implementazione*)