

---

# Palestra di Algoritmi



*Liceo Galilei - Trento*

**#6 - 20/01/2022**

---



# 0. Calendario

**Prossime lezioni:** ONLINE 15-17

- #7 giovedì 27 gennaio 2022
- #8 giovedì 3 febbraio 2022
- OII martedì 8 febbraio 2022

—  
**Siete pronti? Partiamo!**





# Cubetti

## Cubetti colorati

<https://training.olinfo.it/#/task/cubetti/statement>



# Sommelier

## Corso per sommelier

<https://training.olinfo.it/#/task/sommelier/statement>

Suggerimento: backtrack



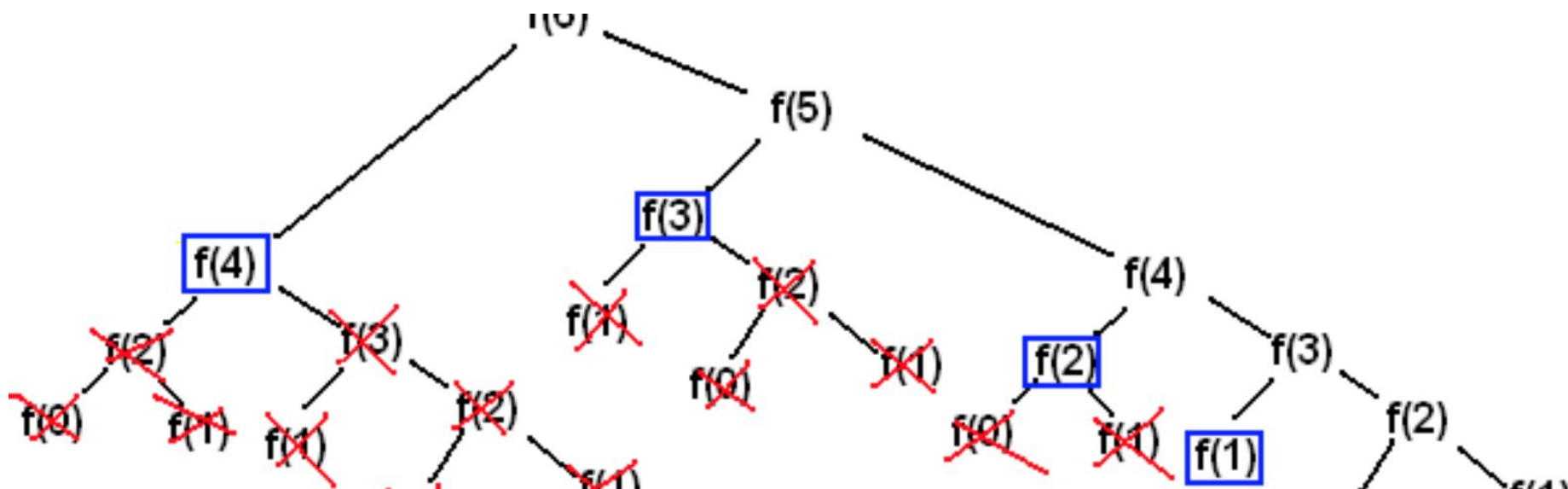
# Discesa

## Discesa massima

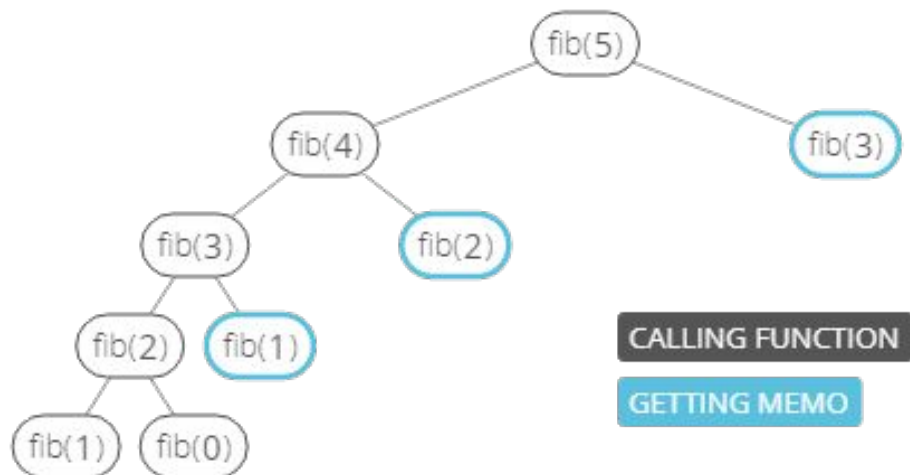
<https://training.olinfo.it/#/task/discesa/statement>

# La tecnica della memoization

Ricordate Fibonacci?



# La tecnica della **memoization**



Calcoliamo **una volta sola** ogni risultato intermedio che ci serve.

Istanziamo un vettore di dimensione  $N$ , dove salvare i risultati calcolati.

Inizializziamo tutto il vettore ad un valore "sentinella" che indica che quel particolare risultato non è stato calcolato, ad es. -1.

Ad ogni passo,

- se  $v[i] == -1$ , allora calcola il risultato, salvalo in  $v[i]$  e ritornalo;
- altrimenti ritorna  $v[i]$



# La tecnica della memoization

```
#include <iostream>

using namespace std;

long long int fib_rec(int n, long long int memo[]){
    if(memo[n] < 0) // devo calcolare il valore
        memo[n] = fib_rec(n-1, memo) + fib_rec(n-2, memo);
    return memo[n];
}

//wrapper
long long int fib(int n) {
    long long int memo[n+1];
    memo[0] = 0;
    memo[1] = 1;
    for(int i=2; i<=n; i++){
        memo[i] = -1;
    }
    return fib_rec(n, memo);
}
```

```
int main(){
    int n = 90;
    cout << fib(n) << endl;
}
```

I tempi di esecuzione sono drasticamente ridotti... ed anche l'utilizzo dello stack

Provare per credere..



SEE YOU  
NEXT TIME!