

Tail recursion

Liceo G.B. Brocchi - Bassano del Grappa (VI)
Liceo Scientifico - opzione scienze applicate
Giovanni Mazzocchin

Tail recursion (ricorsione in coda)

- Esiste una tipologia di ricorsione molto diversa da quella che conosciamo
- D'ora in poi chiameremo la «vecchia» ricorsione **head recursion** (ricorsione in testa)
- La **tail recursion** funziona al contrario rispetto alla **head recursion**

tail recursion	head recursion
operazioni	chiamata ricorsiva
chiamata ricorsiva	operazioni

Tail recursion

- Vediamo come implementare la versione ricorsiva in coda del calcolo del fattoriale
- Per farlo, non partiamo dalla definizione ricorsiva (matematica) di fattoriale, ma dall'implementazione iterativa che conosciamo già
- Iniziamo a scrivere il prototipo della funzione tail recursive:

```
int factorial_it(int num) {  
    int acc = 1;  
    int i = num;  
    while (i >= 1) {  
        acc = acc * i;  
        i--;  
    }  
    return acc;  
}
```

```
int factorial_tr(int num,
```

Tail recursion

- Aggiungiamo la variabile locale `acc` della versione iterativa come parametro della versione tail recursive, ipotizzando che il chiamante lo passi con lo stesso valore (1 nel caso del fattoriale):

```
int factorial_tr(int num, int acc) {
```

Tail recursion

- Trasformiamo la condizione di uscita della versione iterativa in caso base. Portiamo le istruzioni successive al ciclo all'interno del caso base:

```
int factorial_tr(int num, int acc) {  
    if (num == 0) {  
        return acc;  
    }  
}
```

Tail recursion

- Portiamo le istruzioni interne al ciclo nel caso ricorsivo:

```
int factorial_tr(int num, int acc) {  
    if (num == 0) {  
        return acc;  
    }  
    acc = acc * num;  
}
```

Tail recursion

- Invochiamo ricorsivamente la funzione con i parametri esatti:

```
int factorial_tr(int num, int acc) {  
    if (num == 0) {  
        return acc;  
    }  
    acc = acc * num;  
    return factorial_tr(num - 1, acc);  
}
```

Tail recursion

- Con questa versione, i calcoli vengono eseguiti al crescere dello stack (contrariamente a quanto accadeva con la ricorsione classica)
- Lo stack si svuota come al solito, ma svuotandosi non viene effettuato alcun calcolo: viene soltanto restituito il valore finale calcolato

```
int factorial_tr(int num, int acc) {  
    if (num == 0) {  
        return acc;  
    }  
    acc = acc * num;  
    return factorial_tr(num - 1, acc);  
}
```


Da vedere a casa

- [Tail Recursion Explained – Computerphile](#)
- [What on Earth is Recursion? - Computerphile](#)