

## Note su Ricorsione

Riassumo e cerco di chiarire alcuni aspetti lessicali visti in classe relativamente alla programmazione ricorsiva. Per fare ciò, mi appoggio a uno degli esercizi del secondo compito, di cui do quindi una possibile soluzione.

L'esercizio considera il seguente codice C per la funzione di massimo comun divisore (*greatest common divisor*, *gcd*) calcolato secondo l'algoritmo di Euclide:

```
int gcd(int p, int q)
{
    if (q == 0) {return p;}
    return gcd(q, p % q);
}
```

Viene chiesto di:

- 1) indicare il parametro su cui viene fatta la ricorsione e la misura di complessità del problema (dire esplicitamente se la misura coincide col parametro su cui si ricorre), spiegando (brevemente) se e perché questa misura decresce a ogni chiamata ricorsiva.
- 2) Determinare il fattore minimo di decrescita della misura di complessità ad ogni chiamata ricorsiva (es. se la misura è un numero naturale, la decrescita minima potrebbe essere di 1,  $n/2$ , etc... in ciascuna chiamata ricorsiva)

Partiamo con una considerazione generale: una funzione ricorsiva è, per noi, una funzione (ometto i tipi per leggibilità) `fun(p1, ..., pN) {body}` di nome `fun`, parametri (formali) `p1, ..., pN` e corpo `body` in cui il corpo della funzione chiama la funzione `fun` stessa. Per fare ciò, una qualche istruzione in `body` sarà della forma `return fun(q1, ..., qN)`.

Affinché la ricorsione termini, è necessario che ci sia un *fattore di complessità* (questa è l'espressione usata nelle slides su ricorsione (assieme anche a cose tipo "complessità in termini di ..."); in classe avevo usato l'espressione "misura di complessità" come sinonimo) associabile ai parametri e al corpo della funzione (e anche all'ambiente, tecnicamente) che decresce ad ogni chiamata ricorsiva, e si dice sinteticamente che viene fatta ricorsione su quel fattore/rispetto a quella misura.

Nell'esercizio in questione, la domanda viene formulata dando il suggerimento che la misura di complessità sia un parametro della funzione. Infatti, la misura di complessità della funzione è il suo secondo parametro. Vediamo subito che ogni chiamata a `gcd(p, q)` in cui `q` sia diverso da 0, chiama a sua volta `gcd(q, p%q)`. E poiché `p%q` è, per definizione, un numero intero compreso tra 0 e `q-1`, vediamo che in ogni chiamata ricorsiva il secondo parametro della funzione decresce, passando da `q` a `p%q`.

La seconda domanda dell'esercizio chiede invece di determinare il fattore minimo di decrescita della misura di complessità. Qui l'espressione "fattore minimo di decrescita" non è un tecnicismo di C o dei linguaggi di programmazione, ma una normale espressione della lingua italiana che indica il valore numerico che descrive la decrescita della misura di complessità della

funzione (quindi del suo secondo parametro) a ogni chiamata ricorsiva. Ora, poiché in una chiamata ricorsiva passiamo da complessità  $q$  a complessità  $p \% q$ , vediamo che nel peggiore dei casi passeremo da  $q$  a  $q-1$  (infatti  $0 \leq p \% q \leq q-1$ ), e quindi il fattore di decrescita è  $1$ .