La moltiplicazione dei numeri interi

di Cristiano Armellini, cristiano.armellini@alice.it

Moltiplicare due numeri interi è cosa molto semplice. Moltiplicare grandi numeri però non è sempre cosa agevole e studiare algoritmi efficienti è utile al calcolo dei fattoriali e conseguentemente ai sistemi che individuano i numeri primi. Dalle scuole elementari sappiamo che la moltiplicazione equivale sostanzialmente ad un somma,. Così il prodotto 4x3 = 12 si può intendere tanto 3+3+3+3 (somma di quattro volte 3) o 4+4+4 (somma di tre volte quattro) dal momento che vale la proprietà commutativa. Per piccoli numeri questa osservazione ci dice ben poco ma in un calcolatore fa differenza sommare 789 volte 2 o sommare due volte 789 (nel caso si voglia calcolare 789x2). Questa semplice osservazione ci suggerisce un primo algoritmo della moltiplicazione che dati due numeri da sommare prima individua quello maggiore quindi somma tante volte il numero maggiore quante sono le unità del numero minore. Facciamo ora un piccolo passo in avanti se devo moltiplicare due numeri, diciamo **n x e**, posso scrivere:

$$p = ne = n(n + d) = n^2 + nd$$

Ove m è il fattore più piccolo d > 0 è la differenza dei fattori che compongono il prodotto. Calcolare i quadrati di un numero intero è una operazione relativamente veloce così abbiamo individuato un semplice sistema che può accelerare le operazioni i un computer. Infatti il prodotto iniziale si riduce alla somma di un quadrato e di un nuovo prodotto tra numeri più piccoli dove se, necessario, possiamo applicare in via ricorsiva la decomposizione descritta fino a quando non siamo in grado di calcolare facilmente il prodotto. Esempio: $18 * 7 = 7^2 + (18 - 7)7 = 7^2 + 7^2 + (11 - 7)7 = 2 * 7^2 + 28 = 2 * 7^2 + 4^2 + (7 - 4)4$ ecc. Come applicazione a quanto detto scriviamo un applicativo in codice Python (simile al C++) che simula il prodotto di due numeri (con i due algoritmi descritti sopra) e ci fornisce anche il fattoriale di un qualunque numero e i fattoriali di una lista di numeri interi (il fattoriale di un numero, come è ben noto, è il prodotto di tutti gli interi minori o uguali al numero dato n! = n(n-1)(n-2)…..3x2x1.

```
def massimo(a,b):
        if a >= b:
                return a;
        else:
                return b;
def minimo(a,b):
        if a >= b:
                return b;
        else:
                return a;
def prodotto(a,b):
        i = 1;
        m = 0;
        if massimo(a,b) == b:
                while i <= a:
                        m = m+b;
                        i = i+1;
                return m;
        else:
                while i <= b:
```

```
                        m = m+a;
                        i = i+1;
                return m;
def prodotto2(e,f):
        g = minimo(e,f);
        l = massimo(e,f);
        return (prodotto(g,g) + prodotto(l-g,g));
def listafattoriale():
        d = 2;
        k = 3;
        while k < 10:
                d = d*k;
                k = k+1;
                print("-------");
                print(k-1);
                print(d);
                print("-------");
def fattoriale(n):
        j = 2;
        h = 3;
        while h <= n:
                j = j*h;
                h = h+1;
        return j;
```

Un modo molto elegante di calcolare il fattoriale è quello di usare una funzione ricorsiva del tipo:

```
def fattoriale(x):
        if fattoriale < 2:
                return 1;
        else:
                return x*fattoriale(x-1);
```

Mentre in modo del tutto analogo per la potenza di interi abbiamo, sempre in via ricorsiva:

```
def potenza(a,x):
        if x < 1:
                return 1;
        else:
                return a*potenza(a,x-1);
```

Abbiamo già avuto modo di esporre in un precedente articolo che esiste un altro ben più potente e interessante algoritmo che riguarda non solo la moltiplicazione ma anche la somma e la sottrazione di numeri di grandissime dimensioni (anche numeri decimali, non necessariamente interi). Il trucco sta nel considerare i numeri come polinomi per poi ridurli a numeri. Vediamo con un esempio semplice come funziona il meccanismo. Si voglia calcolare 13x 17

13 = 10 +3 = X+3

17 = 10 +7 = X+7

Essendo X = 10

$$(x+3)(x+7) = x^2 + 7x + 3x + 21 = (10 = x) = x^2 + 10x + 2x + 1 = (10 = x) = x^2 + x^2 + 2x + 1$$
$$= 2x^2 + 2x + 1 = 221$$

Ovvero si manipola il polinomio e dove un coefficiente è maggiore di 10 si pone 10=x e si continua nella semplificazione come nell'esempio (una sorta di riduzione modulo 10 dei coefficienti del polinomio in x). Alla fine i coefficienti del polinomio ridotti mi danno il risultato finale. Il sistema funziona anche per la somma e per la sottrazione per numeri con un numero di cifre indefinito.