Manipolazione al calcolatore di serie di potenze

Luca Ferragina

May 21, 2015

1 Introduzione

Lo scopo di questo testo e del codice annesso é quello di costruire un tipo di struttura dati che possa memorizzare le serie di potenze e permetta la gestione delle operazioni fra di esse, in particolare vorremmo che la struttura scelta sia "chiusa per somma e prodotto". Una serie di potenze $\sum_{n=0}^{\infty} a_n x^n$ é univocamente determinata dalla successione $\{a_n\}_{n=0}^{\infty}$, quindi il problema sará, alla fine dei conti, memorizzare la successione. Appare ovvio che per fare ció bisogna trovare un espediente che ci permetta di codificare in maniera finita strutture che di per sé sono infinite come le successioni. I due metodi principali che risolvono questo problema sono i seguenti:

1. Troncare la successione ad un certo intero N.
2. Memorizzare i primi k elementi della successione, $a_0, \ldots, a_{k-1}$ piú una funzione $f(n)$ che agisca da regola per gli elementi succesivi, cioé tale che $a_n = f(n)$ per $n \geq k$.

Entrambi i metodi hanno degli svantaggi: con il secondo metodo esistono successioni non esprimibili, mentre con il primo due successioni che coincidono fino al termine $k - esimo$ ma dopo sono diverse, vengono viste dal calcolatore come la stessa successione. Il metodo che sceglieremo noi é il secondo ed in particolare le funzioni che andremo a considerare sono i polinomi, tuttavia ci sará utile implementare anche il primo metodo. La seguente definizione stabilisce in maniera esatta il tipo di struttura su cui lavoreremo.

Definizione 1.1. Dato un polinomio $p(y) = b_n x^n + \cdots + b_1 x + b_0$ a coefficienti interi, una *serie polinomiale* é una serie di potenze $\sum_{n=0}^{\infty} a_n x^n$ tale che esiste un intero k per cui $a_n = p(n)$ per $n \geq k$.

Osservazione 1.1. In generale il prodotto di due serie polinomiali *non* é una serie polinomiale.

2 Strutture dati e funzioni

2.1 polinomi

Sebbene nel linguaggio C, in cui é scritto tutto il codice, esista una libreria dedicata ai polinomi ne creeremo una nuova (*"poly.h"*) in modo che sia adatta ai nostri scopi. Un *monomio* sará una struttura composta da due interi, che

rappresentano il coefficiente e il grado del monomio, piú un puntatore ad un monomio in modo poi da poter creare liste di monomi. Di conseguenza un polinomio sará definito come un puntatore ad un monomio e verrá implementato come una lista. Non ci soffermeremo molto sulle operazioni fra polinomi che a questo punto saranno piccole modifiche delle classiche operazioni di inserzione e cancellazione sulle liste.

2.2 Serie polinomiali

La struttura *serie polinomiale* é composta da:

- un intero k che che rappresenta il numero di elementi iniziali della serie non rappresentabili attraverso il polinomio;
- un array A di interi di taglia k in cui sono memorizzati i primi k elementi della serie;
- un polinomio $f(x)$.

La funzione che somma due serie polinomiali $S_1 = (a_0, \ldots, a_{k_1-1}, f(x))$ e $S_2 = (b_0, \ldots, b_{k_2-1}, g(x))$ restituisce la serie polinomiale

$$S_3 = (a_0, \ldots, a_{k_1-1}, b_0, \ldots, b_{k_2-1}, f(x) + g(x))$$

2.3 Prodotti di serie polinomiali

Come osservato in precedenza il prodotto di due serie polinomiali non puó essere espresso in generale come una serie polinomiale. Abbiamo quindi bisogno di una nuova struttura, che chiamiamo *prodotto di serie polinomiali*, che raggruppi delle serie polinomiali e che rappresenti appunto il loro prodotto. La struttura *prodotto di serie polinomiali* é composta da:

- un intero m che rappresenta quante sono le serie legate dal prodotto;
- un array di *serie polinomiali* di taglia n.

La funzione che moltiplica due prodotti di serie polinomiali $\Phi_1 = (S_0 \ldots S_{m_1-1})$ e $\Phi_2 = (T_0 \ldots T_{m_2-1})$ restituisce il prodotto di serie polinomiali

$$\Phi_3 = (\Phi_1 \Phi_2) = (S_0 \ldots S_{m_1-1} T_0 \ldots T_{m_2-1})$$

. In particolare si può osservare che la taglia dell'array di Φ_3 é la somma delle taglie dei due array di partenza. L'estensione della struttura appena fatta ci ha permesso di definire il prodotto semplicemente mettendo nella stessa struttura *prodotto* oggetti che non sapevamo moltiplicare, tuttavia questo risultato ci é costato la perdita dell'operazione di somma, infatti è immediato vedere che la somma di due *prodotti di serie polinomiali* non si può esprimere come *prodotto di serie polinomiali*.

2.4 Somme di prodotti

Per ovviare alla perdita dell'operazione somma, ripetiamo lo stesso ragionamento usato in precedenza, ovvero estendiamo nuovamente la struttura definendo la *somma di prodotti*. La struttura *somma di prodotti* è composta da:

- un intero n che rappresenta quante sono le serie legate dal prodotto;
- un array di *prodotti di serie polinomiali* di taglia n.

La funzione che somma due *somme di prodotti* $\Sigma_1 = [\Phi_0 + \cdots + \Phi_{n_1-1}]$ e $\Sigma_2 = [\Psi_0 + \cdots + \Psi_{n_2-1}]$ é molto simile al prodotto precedente e restituisce la *somma di prodotti*

$$\Sigma_3 = [\Phi_0 + \cdots + \Phi_{n_1-1} + \Psi_0 + \cdots + \Psi_{n_2-1}]$$

. Analogamente a prima si ha $n_3 = n_1 + n_2$. Il vantaggio che porta questa nuova struttura é quello di poterne definire anche il prodotto. Grazie alla distributivitá delle serie di potenze, vale infatti:

$$[\Phi_0+\cdots+\Phi_{n_1-1}][\Psi_0+\cdots+\Psi_{n_2-1}] = [(\Phi_0\Psi_0)+\cdots+(\Phi_0\Psi_{n_2-1})+\cdots+(\Phi_{n_1-1}\Psi_0)+\ldots(\Phi_{n_1-1}\Psi_{n_2-1})]$$

3 Serie troncate

Ricapitolando, siamo partiti da un particolare tipo di serie, le *serie polinomiali*, di cui potenzialmente siamo in grado di esprimere l' $n-esimo$ termine della successione per qualsiasi n. Poiché non riuscivamo a compiere alcune operazioni elementari abbiamo esteso due volte la struttura che avevamofino ad ottenere il nostro obiettivo, ovvero una struttura chiusa per somma e prodotto. Tuttavia in questo percorso abbiamo perso la proprietá di conoscere qualsiasi elemento della serie con cui abbiamo a che fare. Per risolvere questo problema quello che faremo é costruire una funzione che prenda in input una *somma di prodotti* e produca in output la corrispondente serie di potenze troncata ad un grado τ fissato.

Osservazione 3.1. La libreria per gestire le serie troncate é stata creata in modo che queste siano rappresentate come array di interi di lunghezza τ e le operazioni siano praticamente uguali alle operazioni sui polinomi.

La funzione che produce la serie troncata corrispondente alla serie polinomiale $S = (a_0, \ldots, a_{k-1}, f(x))$ restituisce come output

$$(a_0, \ldots, a_{k-1}, f(k), \ldots, f(\tau - 1))$$

La funzione che prende in input un prodotto di serie polinomiali e produce in output la serie troncata corrispondente, applica la funzione precedente a tutte le serie polinomiali del prodotto e poi moltiplica fra loro le serie troncate ottenute. La funzione che prende in input una somma di prodotti e produce in output la serie troncata corrispondente, applica la funzione precedente a tutti i prodotti della somma e poi moltiplica fra loro le serie troncate ottenute.