

Olimpiadi di Informatica 2010 Giornate preparatorie

Dipartimento di Informatica
Università di Torino

marzo 2010

5 - Algoritmi su array: il segmento di somma massima.
(versione 29/03/10)

1

3/29/2010

E. Giovannetti -- OI09.

Esercizio preliminare (Programmazione 1)

Dato un array di interi rappresentante una sequenza di temperature giornaliere, trovare la lunghezza del più lungo periodo di gelo, cioè della più lunga sequenza consecutiva di temperature negative o zero.

INVARIANTE:

- la variabile **max** contiene la lunghezza massima di gelo nel periodo $a[0 .. i-1]$
- la variabile **lun** contiene la lunghezza del periodo corrente di gelo, cioè del periodo di gelo che arriva fino ad $a[i - 1]$.



AlgELab-09-10 - Lez. 6

Definizioni preliminari

- *segmento non vuoto* di una sequenza non vuota a_0, a_1, \dots, a_{n-1} è una qualunque sottosequenza a_k, a_{k+1}, \dots, a_h di elementi consecutivi, con $0 \leq k \leq h \leq n-1$;
- in particolare, per una sequenza realizzata come array: *segmento non vuoto* di un array non vuoto a è una qualunque sottosequenza $a[k], a[k+1], \dots, a[h]$ di elementi consecutivi, con $0 \leq k \leq h < a.length$;
- useremo la notazione abbreviata $a[k .. h]$ per indicare il segmento $a[k], a[k+1], \dots, a[h]$ (cioè il segmento da k ad h , estremi compresi); potremo scrivere $a[k]$ invece di $a[k .. k]$;
- useremo talvolta la notazione abbreviata $somma(a[k .. h])$ per indicare la somma degli elementi del segmento, cioè:

$$a[k] + a[k+1] + \dots + a[h]$$

(Nota Bene: tali notazioni NON sono del linguaggio Java).

AlgELab-09-10 - Lez. 6

Problema: ricerca del segmento di somma massima.

Si definisca una procedura la quale, preso un array (non vuoto) di interi, restituisca la somma degli elementi del segmento (non vuoto) di somma massima.

Esempi

array $\{-7, 4, -8, \underline{3, 4, -2, 6}, -10, 1, 3, -9, 9\}$:

il segmento di somma massima è quello sottolineato, e la sua somma è 11;

array $\{\underline{4, 3, 5, 0, 8, 1}\}$

il segmento di somma massima è l'intero array, e la sua somma è 21;

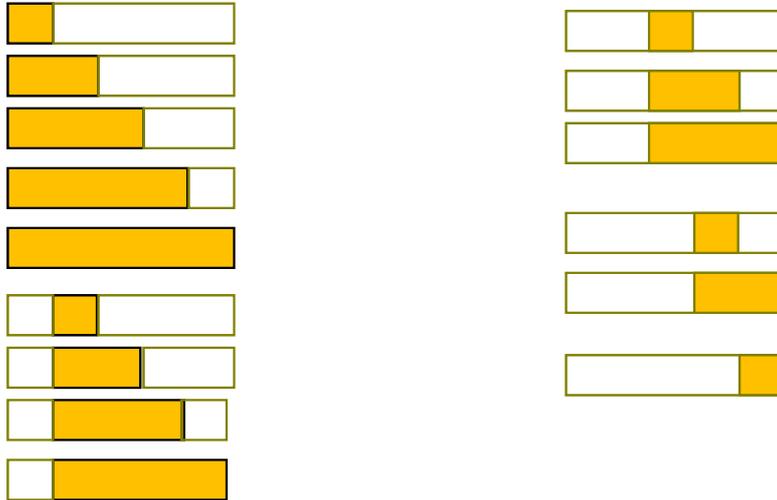
array $\{-4, \underline{-3}, -5, -8\}$

il segmento di somma massima è quello costituito dal solo elemento sottolineato, e la sua somma è -3.

AlgELab-09-10 - Lez. 6

Il segmento di somma massima: soluzione ingenua

- Si calcolano le somme di tutti i possibili segmenti, tenendo memoria della massima.



AlgELab-09-10 - Lez. 6

Il segmento di somma massima

Soluzione naturale:

```
public static int sss(int[] a) {
    int n = a.length;
    int max = a[0];
    for(int i = 0; i < n; i++) {
        calcola la somma di a[i .. j] per ogni j compreso fra i ed n-1
        e confrontala con max, eventualmente aggiornando max;
    }
}
```

Complessità dell'algoritmo (numero di passi)

n iterazioni del ciclo esterno:

prima iterazione:	n iterazioni del ciclo interno;
seconda iterazione:	$n-1$ iterazioni del ciclo interno;
...	
i -esima iterazione:	$n-i$ iterazioni del ciclo interno;

AlgELab-09-10 - Lez. 6

Complessità della soluzione naturale

$$n + (n-1) + (n-2) + \dots + 2 + 1 = n(n+1)/2 = \Theta(n^2)$$

C'è un algoritmo più efficiente e più semplice !
Per trovarlo, occorre scrivere un invariante un po' sofisticato.

AlgELab-09-10 - Lez. 6

Soluzione ottima: l'idea di base

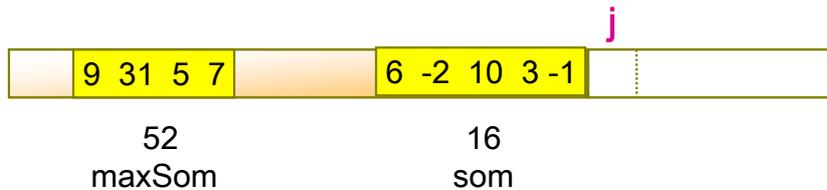
- Si percorre l'array sommando gli elementi successivi in una variabile *som* che contiene quindi la somma corrente, tenendo in una variabile *maxSom* la massima somma trovata fino a quell'istante, e confrontando ad ogni passo *som* con *maxSom* (e aggiornando, se è il caso, *maxSom*).
- Nota: si sommano anche gli interi negativi, perché possono essere seguiti da interi positivi che permettono di raggiungere una somma più grande. Esempio: 3, 4, -2, 1, 3 somma 9.
- Se però diventa negativa la somma corrente, al passo successivo bisogna cominciare una nuova somma corrente, perché ogni somma che contenga il segmento "negativo" sarà minore della corrispondente somma che non lo contiene. Esempio: 3, -2, 4, 6, -15, 7, 3, ... con -15 si ottiene la somma -4; allora si riparte da 7

AlgELab-09-10 - Lez. 6

L'invariante

Nel ciclo si dovranno tenere due variabili contenenti rispettiv.:

- la massima somma contenuta in $a[0 .. j-1]$;
- la massima somma terminante all' indice $j-1$,
o somma corrente.



AlgELab-09-10 - Lez. 6

Il corpo del ciclo, il test, l'inizializzazione

CORPO DEL CICLO:

- aggiorno **som** :

```
if(som > 0) som += a[j];
else som = a[j];
```
- confronto **som** con il massimo **maxSom**,
aggiornando **maxSom** se necessario:

```
if(som > maxSom) maxSom = som;
```
- infine incremento **j**:

```
j++;
```

TEST: `j < a.length;`

INIZIALIZZAZIONE: `som = maxSom = a[0]; j = 1;`

Nota: Invece di un ciclo *while*, si può utilizzare un ciclo *for*.

AlgELab-09-10 - Lez. 6