

1 Film

AlgoFlix è un sistema di streaming online di film. Il sistema memorizza ogni giorno quante persone hanno visto un certo film, usando un vettore P di numeri interi: $P[i]$ è il numero di persone che hanno visto il film nel giorno i . Vogliamo calcolare il massimo numero di giorni consecutivi in cui il numero di persone che hanno visto il film risulta non decrescente. In altre parole, vogliamo calcolare la lunghezza massima tra tutti i sottovettori composti da elementi contigui di P che risultino ordinati in senso non decrescente. Si scriva un algoritmo per risolvere questo problema, se ne determini il costo computazionale e lo si implementi in C.

Note per la consegna. Si scriva il programma in un file di nome `es1-film.c` con il programma. Il file deve contenere un commento con l'analisi della complessità.

Il programma deve leggere da standard input un numero n seguito da una sequenza di n numeri interi, quindi deve stampare la lunghezza del più lungo sottovettore non decrescente.

Esempio di esecuzione 1

Ricevendo da standard input

```
7
1 3 2 5 1 4 7
```

il programma deve stampare

```
3
```

in quanto il più lungo sottovettore non decrescente è composto dagli ultimi tre elementi [1, 4, 7].

Esempio di esecuzione 2

Ricevendo da standard input

```
7
12 3 5 7 8 4 6
```

il programma deve stampare

```
4
```

in quanto il più lungo sottovettore non decrescente è [3, 5, 7, 8] che ha 4 elementi.

Esempio di esecuzione 3

Ricevendo da standard input

```
5
5 4 3 2 1
```

il programma deve stampare

```
1
```

perché gli unici sottovettori non decrescenti sono i singoli elementi del vettore.

2 Lista misteriosa - comprensione e scrittura di codice

Considerate le funzioni `build`, `flist` e `fstring` contenute nel file `es2-lista.c`.

Il tipo `Node` usato nelle funzioni implementa un nodo di una lista concatenata; questo tipo, come d'abitudine, ha due membri: un membro che contiene le informazioni relative all'elemento della lista e un membro che punta al prossimo elemento della lista.

Comprensione del codice. Esaminate il codice e rispondete alle seguenti domande.

1. Descrivete a parole il comportamento della funzione `build`. Cosa fa e come lo fa? In particolare descrivete:
 - a) cosa restituisce la funzione `build`;
 - b) quante volte viene invocata la funzione `flist`;
 - c) cosa viene verificato nell'`if` a riga 9
 - d) cosa fa l'assegnamento nella riga 12
2. Riassumete con una frase il comportamento della funzione `fstring`. Cosa fa e come lo fa? In particolare, come deve essere il primo argomento che si passa a tale funzione?

Scrittura di codice.

1. Definite, in C, un tipo `Node` coerentemente a come viene usato nelle funzioni.
2. Scrivete una funzione `fstring_iter` che produca lo stesso risultato di `fstring` ma senza usare la ricorsione e senza usare le funzioni di `string.h`.
3. Partendo dalle funzioni `build` e `flist`, scrivete delle varianti `build2` e `flist2` in modo che le istruzioni

```
char word[20] = "";  
printf( "%s\n", fstring( parola, build2( "hello" ) ) );
```

producano in output la stringa
hellohellhelheh

Note per la consegna. Salvate il file allegato con nome `es2-listasvolto.c`. Scrivete le risposte alle domande di comprensione del codice in un commento all'inizio del file. Completate il file come specificato sopra: il file dovrà contenere, oltre alle funzioni già fornite, anche le funzioni `fstring_iter`, `build2` e `flist2`. Il programma consegnato **non deve contenere** alcuna funzione `main` (anche se naturalmente è opportuno fare dei test durante la scrittura del programma!!!)