

Laboratorio 3 — Analizzatore Genetico con Mutazioni Casuali

Assignment — dna_mutazioni.c

Università di Catania — Dipartimento di Matematica e Informatica

Corso di Laurea in Informatica

Programmazione 1 e Laboratorio (Canale O-Z)

Descrizione del Programma

Si scriva un programma in C, chiamato `dna_mutazioni.c`, che analizzi una sequenza numerica di espressione genetica. La sequenza iniziale viene acquisita da input e poi perturbata con un numero prefissato di mutazioni pseudo-casuali. L'obiettivo finale è individuare la sottosequenza contigua strettamente crescente più lunga nella sequenza mutata.

Sia $G[i]$ il livello di espressione del gene in posizione i , con $0 \leq i < N$. Per ogni mutazione si sceglie un indice casuale idx e una variazione intera Δ in $[-5, 5]$:

$$G[idx]_{nuovo} = \max(0, G[idx]_{vecchio} + \Delta)$$

Data una sottosequenza contigua crescente da a a b , la sua lunghezza vale:

$$L = b - a + 1$$

Il programma deve stampare risultati intermedi e statistiche finali, restituendo opportuni codici di terminazione.

Requisiti di Implementazione

PUNTO A: Leggere da standard input i parametri: N (numero geni), M (numero mutazioni), `seed` (seed pseudo-casuale).

- Stampare un unico prompt (ad esempio: `Inserisci N, M, seed:`).
- Vincoli: $1 \leq N \leq 200$, $0 \leq M \leq 5000$, $seed \geq 0$.
- In caso di parametri non validi stampare `Errore: parametri genetici non validi.` e terminare con exit status 1.

PUNTO B: Dichiarare un array statico `int geni[200]`; e leggere i N valori iniziali di espressione da input.

- Stampare il prompt (ad esempio: `Inserisci i N livelli iniziali:`).
- Vincolo su ogni valore: $0 \leq G[i] \leq 1000$.
- Se almeno un valore non e' valido, stampare `Errore: livello genico fuori intervallo.` e terminare con exit status 1.

PUNTO C: Stampare la sequenza iniziale, inizializzare il generatore con `srand(seed)` e applicare M mutazioni con un ciclo `for`:

- `idx = rand()%N`
- `$\Delta = (rand()\%11) - 5$`
- aggiornare `geni[idx]` imponendo limite inferiore a 0.

Durante il ciclo, contare quante mutazioni sono: `incremento` ($\Delta > 0$), `decremento` ($\Delta < 0$), `nulle` ($\Delta = 0$).

PUNTO D: Terminata la fase di mutazione, stampare la sequenza finale e cercare la sottosequenza contigua strettamente crescente piu' lunga. In caso di parita' di lunghezza, scegliere la prima occorrenza. Stampare:

- indice iniziale e finale della run migliore;
- lunghezza massima L_{max} .

PUNTO E: Calcolare sulla run migliore:

- media (`double`);
- minimo;
- massimo.

Stampare inoltre il riepilogo mutazioni (incremento/decremento/-nulle) e terminare con exit status 0.

Output di Controllo (Esempi di Esecuzione)

Caso 1 — Caso valido con mutazioni e run crescente significativa

Inserendo i dati:

```
8 12 12345
10 12 9 11 15 14 20 18
```

Il programma deve stampare a schermo (valori numerici indicativi):

```
Inserisci N, M, seed: 8 12 12345
Inserisci i N livelli iniziali: 10 12 9 11 15 14 20 18
Sequenza iniziale: 10 12 9 11 15 14 20 18
Sequenza mutata: 10 13 7 12 15 14 23 16
Mutazioni - incremento: 5, decremento: 6, nulle: 1
Run crescente migliore: start=2 end=4 lunghezza=3
Statistiche run migliore - media: 11.33 min: 7 max: 15
```

(Exit code atteso: 0)

Caso 2 — Parametro iniziale non valido

Inserendo i dati:

```
0 10 99
```

Il programma deve stampare a schermo:

```
Inserisci N, M, seed: 0 10 99
Errore: parametri genetici non validi.
```

(Exit code atteso: 1)

Caso 3 — Valore genico fuori intervallo

Inserendo i dati:

```
5 3 7
12 14 2000 10 9
```

Il programma deve stampare a schermo:

```
Inserisci N, M, seed: 5 3 7
Inserisci i N livelli iniziali: 12 14 2000 10 9
Errore: livello genico fuori intervallo.
```

(Exit code atteso: 1)