

Esonero a.a. 2018-2019

Programmazione I con Laboratorio

7 Novembre 2018

Preambolo. Il vecchio zio d’America, *Rosco*, è passato a miglior vita dopo essere diventato ricchissimo in vita comprando e vendendo azioni a *Wall Street*. Rosco è stato però anche famoso per essere avarissimo e non aver mai aiutato chi aveva bisogno. Per questo motivo, per poter scontare i suoi peccati fino in fondo, è costretto ad apparire in sogno ed aiutare i più poveri predicando il futuro delle azioni della *Toppe e Rammendi spa*, in modo che anche altri possano arricchirsi come ha fatto lui in vita.

Testo. Rosco ogni volta può predire il futuro di n giorni del valore delle azioni in questione. Per esempio, la Figura 1 riporta un esempio di quello che Rosco può rivelare al prossimo fortunato: al giorno 0 (il giorno dell’apparizione), le azioni della Toppe e Rammendi spa valgono 10, al giorno 1 valgono 14, e così via. Per aiutare Rosco hai bisogno di creare un programma in linguaggio *C* che trovi il giorno migliore in cui comprare ed il giorno migliore in cui vendere le azioni massimizzando il guadagno, dato l’intervallo di n giorni.

Notare che il giorno 0 non si può comprare le azioni: Rosco appare di notte quando il mercato è già chiuso, quindi i valori in Figura 1 corrispondono ai valori di chiusura per ogni giorno. Se si compra all’inizio del giorno 1 si guadagna 4 a fine giornata (le azioni passano da 10 a 14).

Scrivere una funzione *main* che:

- Legga da tastiera il numero di giorni n della predizione, ed il numero totale k di azioni comprate.
- Crei un array di lunghezza n e legga da tastiera il valore delle azioni per ogni giorno (per esempio, i 12 valori dell’asse x in Figura 1).
- Crei un array di $n - 1$ posizioni e lo riempi considerando la differenza tra il valore del giorno precedente e successivo. Per l’esempio in Figura 1 il nuovo array avrà valori $\{+4, -6, +3, +1, +3, -2, +3, -4, +1, -9, +6\}$.
- Trovi l’intervallo migliore su questo nuovo array, cioè il giorno in cui comprare e il giorno in cui vendere per poter massimizzare il guadagno dato il valore di una azione. Per l’esempio in Figura 1 l’intervallo migliore consiste nel comprare il giorno 3 e vendere il giorno 7 (2-6 suol nuovo array di $n - 1$ posizioni). Questi giorni devono essere stampati sullo schermo in

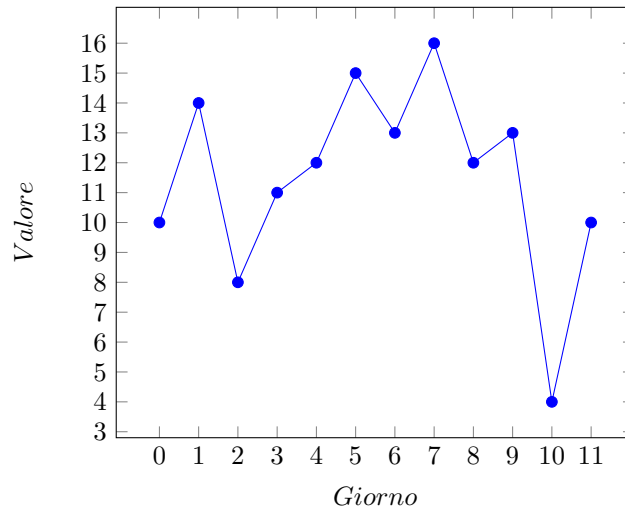


Figure 1: Una possibile predizione di Rosco.

modo da suggerire a Rosco la soluzione.

(Sugg.: l'intervallo migliore è quello a somma massima sul secondo array)

- Nel caso particolare in cui non esista un intervallo possibile in cui si guadagni, consigli di non comprare affatto le azioni.
- Infine stampi l'ammontare totale riscosso dalla vendita ed il guadagno (in base a k). Per l'esempio in Figura 1, comprando a 8 dollari e rivendendo a 16, se $k = 500$, l'ammontare riscosso è 8000 ed il guadagno è 4000.

Se esistono più intervalli di costo (massimo) uguale, è sufficiente ritornare come soluzione uno solo di essi.

Vedere i prezzi delle azioni sul grafico in Figura 1 come i prezzi di chiusura del giorno corrispondente. Se si compra ad un dato giorno, si compra al prezzo di chiusura del giorno prima (in pratica è come se si comprasse appena il mercato apre, ed il prezzo è sempre quello del giorno prima) Se si vende ad un dato giorno si vende al prezzo di chiusura di quel giorno. In realtà è semplicemente l'intervallo con la differenza massima tra giorno finale e giorno iniziale, ricordandosi però di incrementare di 1 l'indice del giorno in cui si compra. Rispetto alla Figura 1, se si compra e vende il giorno 3 si guadagna +3: si compra ad 8 e si vende a 11.

Consegna. Consegnare utilizzando GitHub, seguendo l'indirizzo: <https://classroom.github.com/a/UswJOAem> entro Venerdì 16 Novembre alle ore 23:59. Leggere il file *README.md* del progetto e seguire le istruzioni.