



Università degli Studi dell'Insubria
Dipartimento di Scienze Teoriche e Applicate

Programmazione Concorrente e Distribuita Soluzioni che sfruttano il parallelismo: esercizi

Luigi Lavazza
Dipartimento di Scienze Teoriche e Applicate
luigi.lavazza@uninsubria.it



Esercizio 1: somma righe matrice

- Sia A una matrice di dimensione $n \times m$.
- Si vuole ottenere il vettore C i cui elementi sono definiti come segue:

$$c_i = \sum_{k=0}^{m-1} a_{ik}$$

dove $0 \leq i < n$

- Implementare un programma concorrente per calcolare gli elementi c_i in parallelo.



Esercizio 1: somma righe matrice

- Dobbiamo calcolare, per ogni i

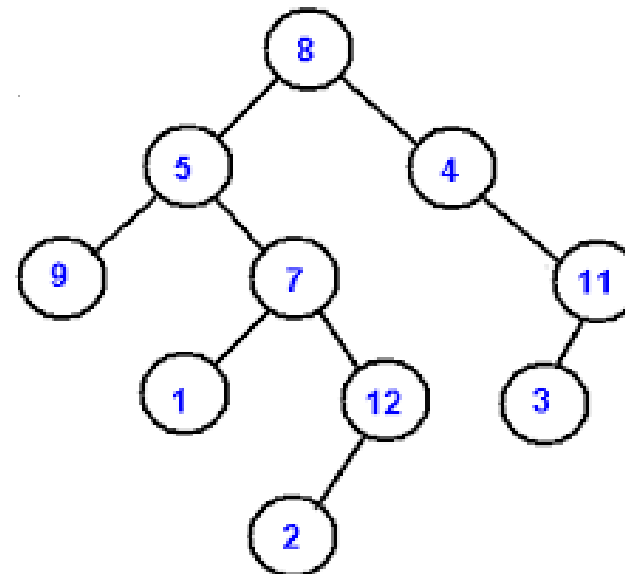
$$c_i = \sum_{k=0}^{m-1} a_{ik}$$

- Un programma sequenziale fa prima tutti i calcoli per $i=0$, poi tutti i calcoli per $i=1$, ecc...
- Un programma multithread può creare un thread che si occupa di c_0 , un altro che si occupa di c_1 , uno per c_2 , ecc.



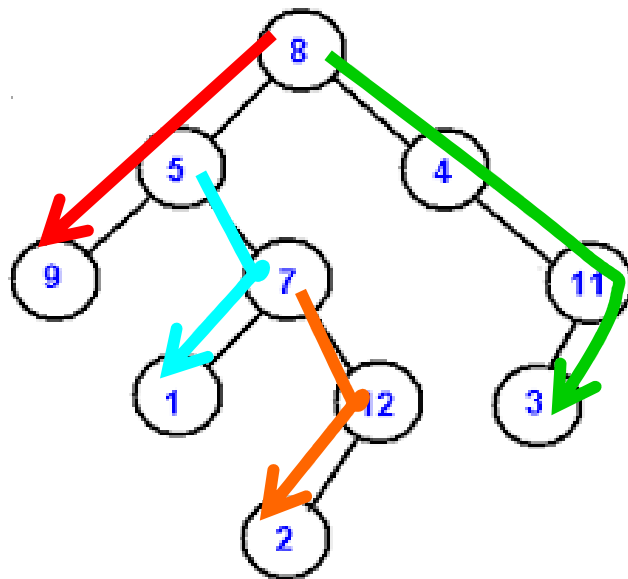
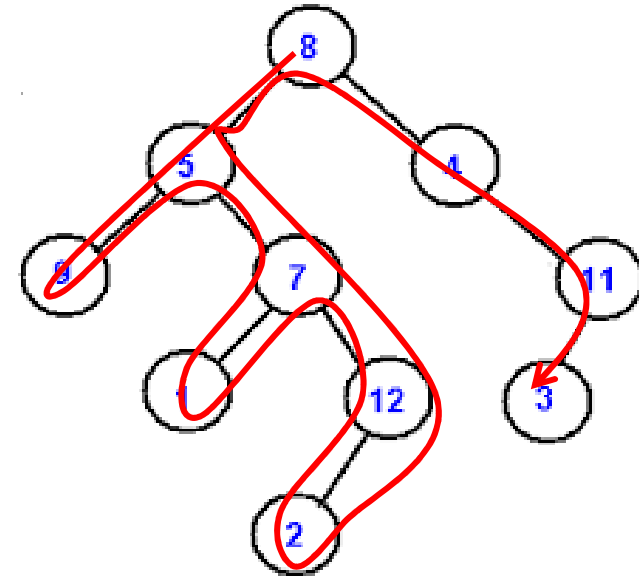
Esercizio 2: ricerca in alberi binari

- Scrivere un programma che sfrutta molteplici thread per effettuare una ricerca in un albero binario dato.



Esercizio 2: ricerca in alberi binari

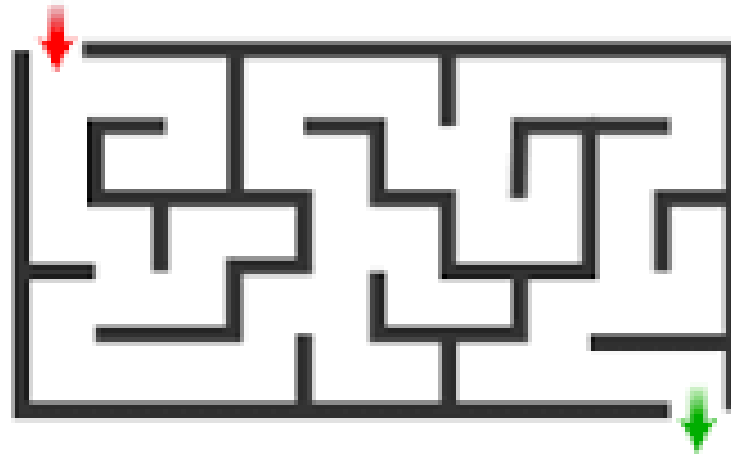
- Suggerimento.
- Un programma sequenziale visita tutto l'albero.



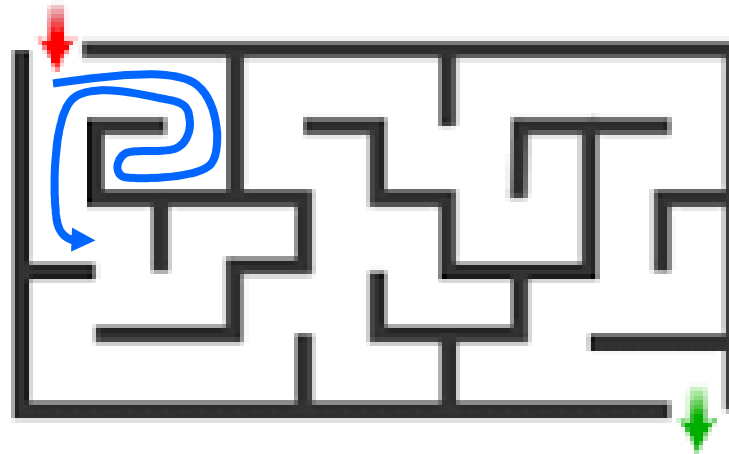
- Un programma multithread può far visitare ciascuna «sezione» dell'albero da un thread dedicato.



Esercizio 3: trovare la strada in un labirinto



Esercizio 3: trovare la strada in un labirinto

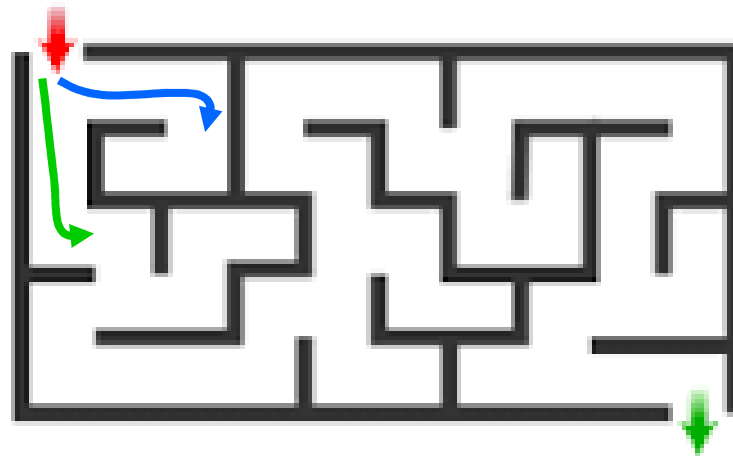


- Un programma tradizionale quando trova un bivio deve:
 - ▶ Esplorare il primo ramo
 - ▶ Se non ha trovato l'uscita, tornare indietro ed esplorare il ramo alternativo



Esercizio 3: trovare la strada in un labirinto

- Un programma multithread quando trova un bivio può proseguire da una parte e generare un thread che esplori le strade alternative



- NB: quando un thread arriva in un posto già visitato da un altro thread può terminare.