

Esercizi 28-11-23

Riccardo Gennaro

November 2023

Esercizio 1 (Coda a priorità con liste)

Un messaggio viene rappresentato mediante una struct

```
1 struct messaggio {  
2     char testo[10000];  
3     int priority;  
4 }
```

dove priorità ha un range crescente da 1 a 10. Scrivere una struttura dati "coda a priorità" (con una lista di nodi) in cui e' possibile inserire messaggi, da cui vengono estratti in modo FIFO per classi di priorità: (prima quelli a priorità 10, poi quelli a priorità 9,...)

Esercizio 2 (Allocazione dinamica e matrici)

Scrivere un programma che calcoli il *prodotto matriciale* di due matrici di numeri interi. Se le matrici non sono compatibili per il prodotto stampare un messaggio di errore.

La dimensione delle matrici deve essere specificata dall'utente (i.e. usare l'allocazione dinamica). Il prodotto è calcolato con la funzione seguente:

```
1 int** compute_prod(int** m1, int r1, int c1, int** m2, int c2);
```

Dove

- m1 è la prima matrice;
- r1 è il numero di righe di m1;
- c1 è il numero di colonne di m1;
- m2 è la seconda matrice;

- `c2` è il numero di colonne di `m2`;

NOTA: la complessità di `compute_prod(...)` deve essere pari o inferiore a $O(n^3)$. Ciò significa che potete implementare la funzione con l'algoritmo che si basa sulla definizione del prodotto fra matrici, oppure, se siete dei pazzi maniaci, potete implementare l'[algoritmo di Strassen](#).