Pile e Code

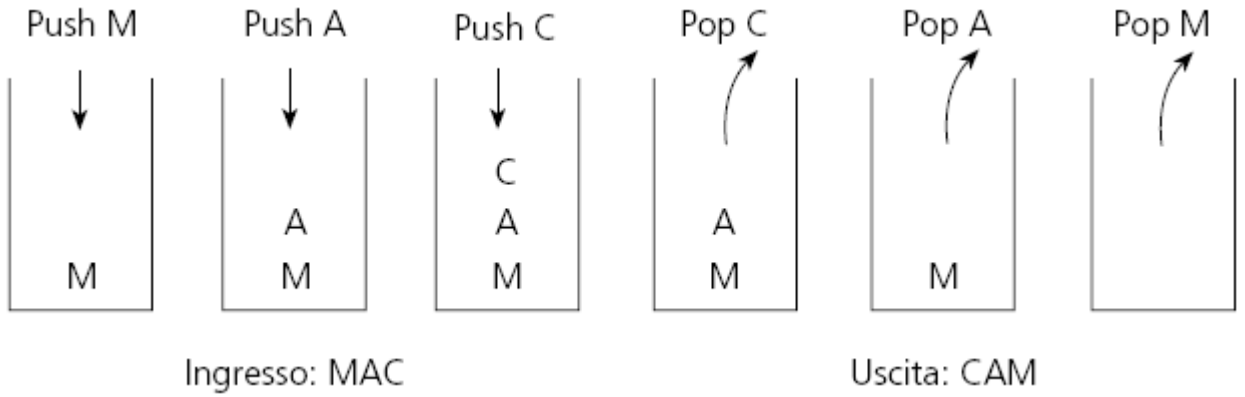
Alessandro Ortis

ortis@dmi.unict.it
www.dmi.unict.it/ortis/

Pila

- uno *stack* è una lista alla quale si può accedere solo da un estremo

- la pila è gestita con strategia Last In First Out (LIFO), cioè la sequenza di estrazione di elementi dallo stack è esattamente l'inversa di quella di immissione



- le operazioni usuali nella pila sono **Inserire** (push), per aggiungere un elemento alla cima della pila e **Rimuovere** (pop) per eliminare un elemento dalla cima della pila

Pila

- Una pila può essere implementata tramite array
  - In tal caso la sua dimensione è fissata
- O tramite puntatori e liste concatenate
  - Non vi sono restrizioni alla sua dimensione

- Una pila può essere vuota o piena
  - Se proviamo ad estrarre elementi da una pila vuota si verificherà un errore di underflow

Implementare pile con Array

- Definibile via classi
- I suoi membri includeranno (oltre all'array)
  - Indice che punta alla cima della pila
  - Insieme di operazioni sulla pila.
- La dimensione della pila non può superare il numero di elementi dell'array.

- Il fondo della pila è la posizione 0 dell'array.
- Il primo elemento è inserito in posizione 0, poi in posizione 1, etc.
- L'indice che segna la cima della pila si incrementa di 1 ogni volta che un nuovo elemento è inserito

Implementare pile con Array

Push

- Verificare che la pila non sia piena
- Incrementare di 1 la cima della pila
- Inserire l'elemento in cima

Pop

- Verificare che la pila non sia vuota
- Leggere l'elemento in cima alla pila
- Ridurre di 1 la cima

Implementare pile con liste dinamiche

- Basta una lista semplice che non sarà mai piena (a meno che non vogliamo stabilire noi un limite al numero di elementi).
- La pila è vuota se il puntatore alla cima è NULL

Push

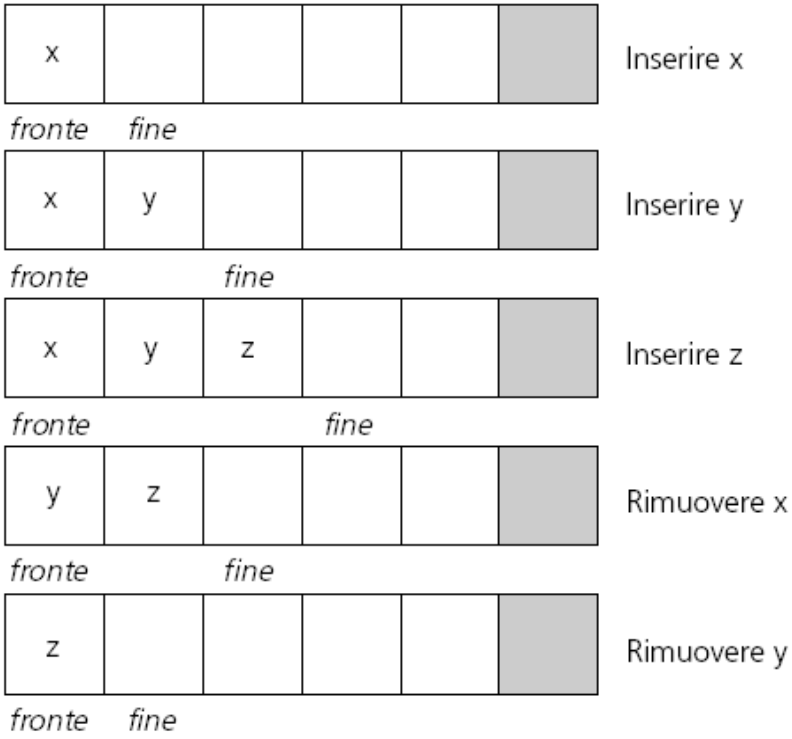
- Aggiunge un nuovo nodo, in testa alla lista, con il dato che si desidera aggiungere, come primo elemento

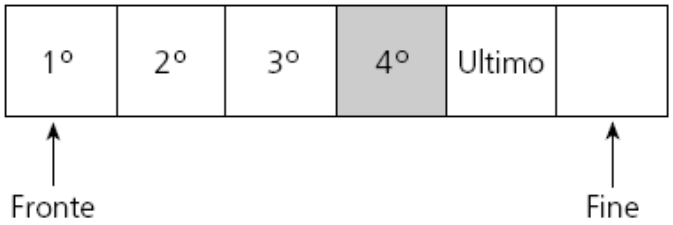
Pop

- Verifica che la lista non sia vuota
- Estrae il valore del primo nodo.
- Cancella il primo nodo.

Concetto e gestione di una coda

La **coda** (queue) è una struttura dati gestita con politica *First In First Out* (FIFO)





permette di accedere ai dati da ciascuno dei suoi due estremi; gli elementi s'immettono in coda e si rimuovono dalla testa nello stesso ordine in cui sono stati immessi

Operazioni su code

- **enqueue, front, dequeue, isEmpty, isFull**

- **enqueue** –aggiunge un elemento in coda.
- **dequeue** – rimuove l'elemento in testa alla coda.
- **front** – legge l'elemento in testa alla coda senza rimuoverlo.
- **isEmpty** – restituisce true se la coda è vuota.
- **IsFull** – restituisce true se la pila è piena (nel caso l'implementazione preveda una capacità massima)

Realizzazione con Array

La realizzazione con array prevede:

- l'uso di un array per memorizzare gli elementi nella coda;
- Un *int* che memorizza la dimensione massima dell'array
- Un indice che individua l'elemento attualmente in testa alla coda
- Un indice che individua l'elemento attualmente alla fine della coda

Realizzazione con Array: idea di base

- Invece di spostare gli elementi dell'array modifichiamo solo gli indici che puntano alla testa e alla fine
- Inizializziamo i due interi testa =0 e fine =-1
- Ogni volta che inseriamo un elemento spostiamo fine (fine++)
- Ogni volta che estraiamo un elemento spostiamo testa (testa++)

Problema:

- Il procedimento funziona bene solo fino a quando fine non raggiunge la fine dell'array.

Soluzione:

Array Circolari!

Esercizi

Implementare:

- una classe pila usando un array
- una classe coda usando un array
- una classe pila per tipi generici
- una classe pila dinamica
- una classe coda dinamica