



UNIVERSITÀ
DI TRENTO

Dipartimento di Ingegneria e
Scienza dell'Informazione
DISI - Trento

Programmazione 1

20 - Esercitazione

Matteo Franzil

matteo.franzil@unitn.it

Anno Accademico 2023/2024

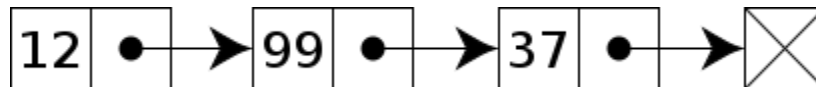
Nelle puntate precedenti

liste concatenate

Una lista concatenata `nodo` di oggetti di tipo `T` è una `struct` contenente:

1. un campo `value` di tipo `T`
2. un campo `next` di tipo `*nodo`

```
struct nodo {  
    T value;  
    nodo *next;  
};
```



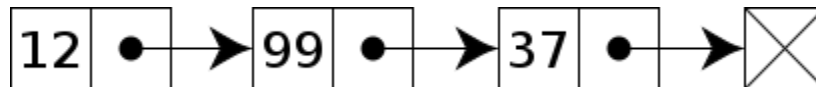
Nelle puntate precedenti

liste concatenate

Una lista concatenata `nodo` di oggetti di tipo `T` è una `struct` contenente:

1. un campo `value` di tipo `T`
2. un campo `next` di tipo `lista` (alias per `*nodo`)

```
struct nodo;  
typedef nodo* lista;  
struct nodo {  
    T value;  
    lista next;  
};
```



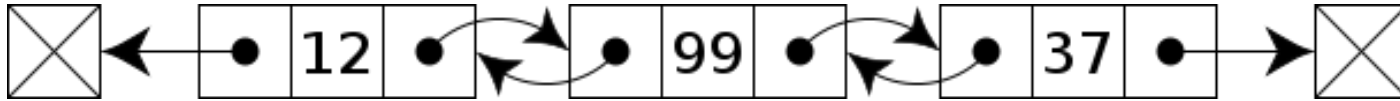
Nelle puntate precedenti

pros&cons

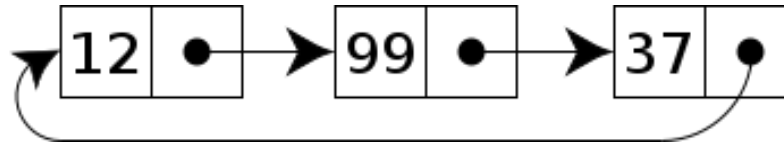
- inserimento e rimozione più efficiente rispetto ad array
- liste concatenate hanno dimensione realmente dinamica
- ricerca meno efficiente rispetto ad array
- usano più memoria rispetto ad array (puntatori)
- accesso solo sequenziale
- allocazione non contigua (efficienza cache CPU ridotta)
- ...

Nelle puntate precedenti

varianti



lista concatenata doppia



lista circolare

Nelle puntate precedenti

utilizzo

Utilizzare per implementare strutture dati astratte quali

- pile
- code
- alberi
- ...

00 - TDA

Definizione

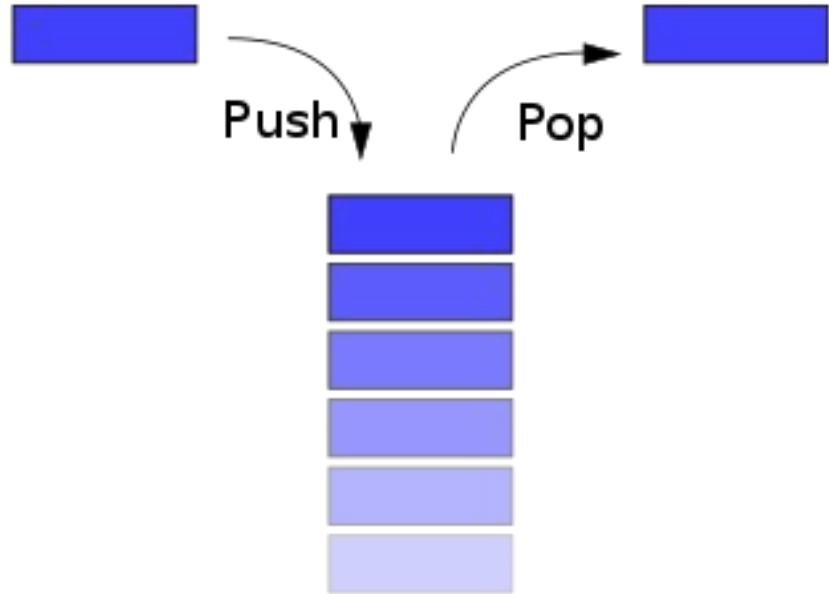
Un insieme di valori (variabili) e di operazioni (funzioni) definite indipendentemente dalla loro implementazione

- Pile (Stack)
- Code (Queue)
- Alberi (Tree)
- ...

00 - TDA

Pila

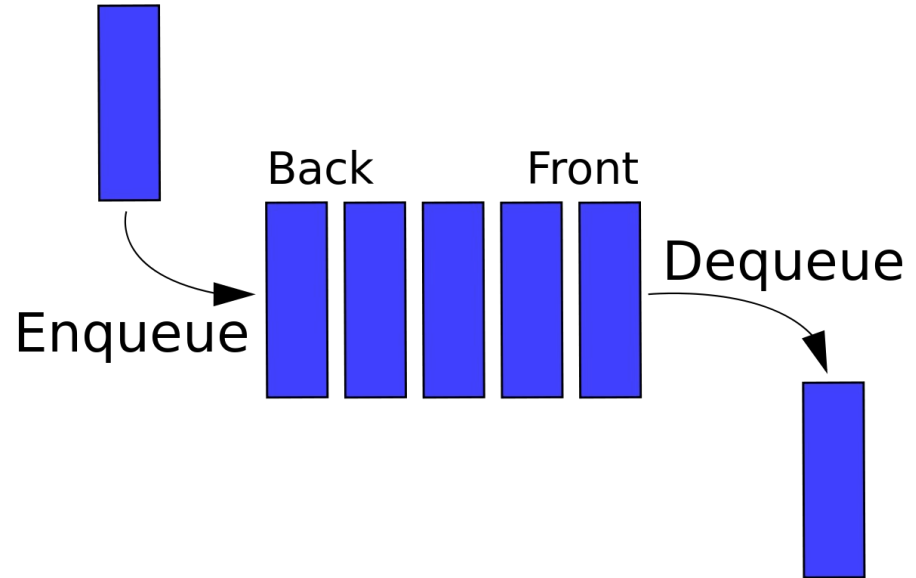
- `init()`
- `deinit()`
- `push(T)`
- `pop()`
- `top(&T)`



00 - TDA

Coda

- `init()`
- `deinit()`
- `enqueue(T)`
- `dequeue()`
- `first(&T)`



01 - Inverti Righe

Partendo dall'implementazione di una Pila tramite liste concatenate
<https://pastebin.com/BCyD0f4g> (.cc), <https://pastebin.com/Rj4UhUA0> (.h)

scrivere un programma che prenda in ingresso un file con un intero per riga e lo copi su un altro file con le righe in ordine inverso

1		3
2	→	2
3		1

02 - Controllo Formula

Partendo dall'implementazione di una Pila tramite liste concatenate
<https://pastebin.com/BCyD0f4g> (.cc), <https://pastebin.com/Rj4UhUA0> (.h)

scrivere un programma che legge una sequenza di caratteri in input
dall'utente e determina se le parentesi sono “bilanciate”:

((() ())) => **SI**

() () (() => **NO**

03 - Alle Poste

**Partendo dall'implementazione di una Coda tramite liste concatenate
con tipo di dato `char*`**

<https://pastebin.com/yuqZUM5j> (.cc), <https://pastebin.com/TNvTjBDU> (.h)

**Scrivere un programma che simuli l'arrivo e lo smaltimento di una coda all'ufficio postale. Gli elementi della coda sono i nomi delle persone. Il main sarà un menù con tre opzioni: (1) aggiungi persona, (2) smaltisci persona, (3) stampa situazione corrente.
Non sono ammessi omonimi nella coda.**

Potete usare la funzione `strcmp` della libreria `<cstring>`

Per maggiori informazioni:

<http://www.cplusplus.com/reference/cstring/strcmp/>

04 - Inverti Pila

coda (int): <https://pastebin.com/SdxreBUc> (.cc) <https://pastebin.com/z0J4t7YY> (.h)
pila (int): <https://pastebin.com/BCyD0f4g> (.cc) <https://pastebin.com/Rj4UhUA0> (.h)

Scrivere un programma che inverta gli elementi di una pila usando una coda.

1		3
2	→	2
3		1

05 - Palindroma

coda: <https://pastebin.com/SdxreBUc> (.cc) <https://pastebin.com/z0J4t7YY> (.h)
pila: <https://pastebin.com/BCyD0f4g> (.cc) <https://pastebin.com/Rj4UhUA0> (.h)

Scrivere un programma che, data una parola in input (massimo 100 caratteri), dica se è palindroma oppure no usando una pila e/o una coda.

**Nota: non potete sfruttare la conoscenza della lunghezza della parola
(non potete nemmeno calcolarla)**