

10. Strutture Dati

Corso di Algoritmi e Linguaggi di Programmazione Python/C

Outline

- L'importanza delle strutture dati
- Array
- Liste
- Struct ed Union
- Pile e Code

L'importanza delle strutture dati

- Le strutture dati permettono di contenere dati più complessi di un dato di tipo primitivo.
- Senza una struttura dati, l'unico modo di rappresentare un insieme di n numeri è quello di creare n :

```
int primo_numero = 1;  
int secondo_numero = 2;  
...  
int ennesimo_numero = n;
```

- All'aumentare di n , ovviamente, questa soluzione diventa decisamente più complessa da attuare.

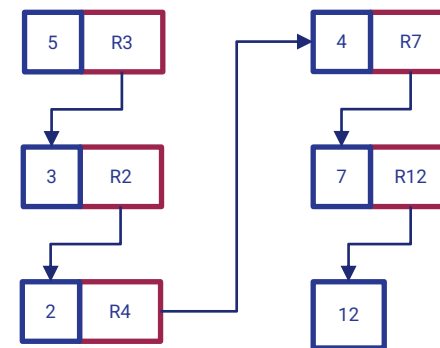
Array

- Un **array** è concettualmente riconducibile ad un vettore
- L'indicizzazione parte da 0 e termina ad $n - 1$, con n lunghezza dell'array.
 - *Un'eccezione notevole è MATLAB, in cui gli indici partono invece da 1.*
- L'array ha lunghezza predefinita.
- L'array è una struttura ad **accesso diretto**: infatti, possiamo usare un indice per accedere ad un ben determinato elemento.

valore	8	5	12	7	4
indice	0	1	2	3	4

Liste

- Simili agli array, ma con una differenza fondamentale.
- Ogni elemento contiene un riferimento all'elemento successivo.
- La lista è una struttura ad **accesso sequenziale**: per accedere agli elementi non si usa un indice, ma un apposito costrutto chiamato **iteratore** (o, in alternativa, un ciclo).
- Risulta anche più semplice da modificare di un semplice array: infatti, non dobbiamo ricostruire l'intero array, ma ci basta cambiare i riferimenti degli elementi immediatamente precedenti e successivi dell'elemento da inserire o cancellare.



Struct ed union

- **Struct**

- Contiene un *insieme* di dati di natura eterogenea.

```
struct nome_struct {  
    tipo_campo_uno id_campo_uno;  
    tipo_campo_due id_campo_due;  
};
```

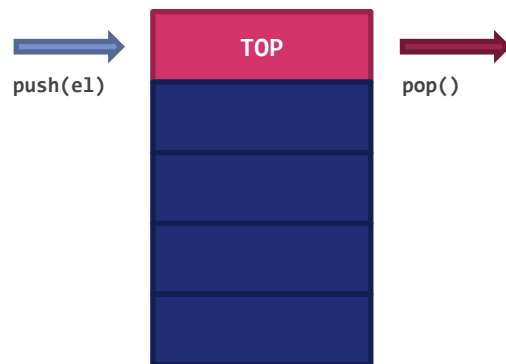
- **Union**

- Simile alla **struct**, ma con la differenza che il dato rappresentato può essere scelto da un ventaglio di tipi.

```
union nome_union {  
    tipo_union_uno id_union_tipo_uno;  
    tipo_union_due id_union_tipo_due;  
};
```

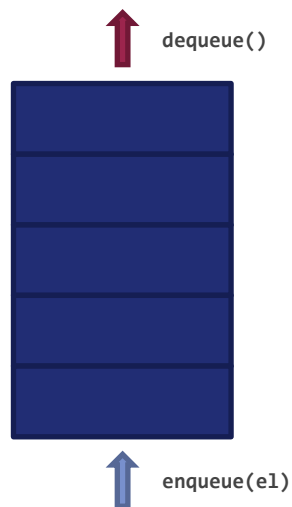
Pile e code

- Strutture ad accesso **limitato**
 - *Inserzione e rimozione di un dato alla volta*
- **Pile**: sfruttano il paradigma **LIFO (Last-In, First-Out)**
 - *L'ultimo ad accedere è il primo ad uscire*



Pile e code

- **Code:** sfruttano il paradigma **FIFO (First-In, First-Out)**
 - *Il primo ad accedere è il primo ad uscire*
 - *Ricordano le code cui siamo normalmente abituati*



Domande?

42