

Strutture ricorsive

- **Ciclo**: iterazione di una sequenza di passi attraverso esecuzioni ripetute sugli **stessi dati**
- **Ricorsione**: ripetizione di tutte le istruzioni come sottocompito su **dati parziali**
- Esempio: foto di una foto di una foto...

Esempio: algoritmo di ricerca binaria (in un insieme ordinato)

- Tecnica che usiamo spesso per cercare una voce in un dizionario:
 - Apriamo il dizionario in un punto (a meta')
 - Se non c'e' la voce che cerchiamo, andiamo nella prima parte o nella seconda
- Finche'
 - Troviamo la voce cercata → si
 - Guardiamo un pezzo con un elemento singolo e non e' la voce cercata → no

Esempio

In una lista ordinata di nomi cerchiamo **John**

Original list	First sublist	Second sublist
Alice Bob Carol David Elaine Fred George Harry Irene John Kelly Larry Mary Nancy Oliver	Irene John Kelly Larry Mary Nancy Oliver	Irene John Kelly

Pseudocodice 1

```
if (lista vuota)
  then fallimento
else
  (elemento-test ← elemento-a-meta';
  Scelta fra tre casi:
  Caso 1: elemento-cercato = elemento-test
          (successo)
  Caso 2: elemento-cercato < elemento-test
          (cerca nella parte prima di elemento-test)
  Caso 3: elemento-cercato > elemento-test
          (cerca nella parte dopo elemento-test)
  )
```

Pseudocodice 2: nome della procedura

Procedure Ricerca(lista, elemento-cercato)

if (lista vuota)

then fallimento

else

 (elemento-test \leftarrow elemento-a-meta';

 Scelta fra tre casi:

Caso 1: elemento-cercato = elemento-test
 (successo)

Caso 2: elemento-cercato < elemento-test
 (**Ricerca**(parte prima di elemento-test,ec))

Caso 3: elemento-cercato > elemento-test
 (**Ricerca**(parte dopo elemento-test,ec))

)

Esempio 1

- Cerchiamo **Bill** nella lista
(Alice, Bill, Carol, David, Evelyn, Fred, George)
- Seleziono David
- David \neq Bill \rightarrow cerco nella prima meta' (Alice, Bill, Carol)
 - Sospendo l'esecuzione di Ricerca in corso, e attivo un'altra esecuzione della procedura Ricerca
 - Seleziono Bill
 - Bill = Bill \rightarrow si
- Ritorno nella prima esecuzione \rightarrow si

Riassunto

- Divide la lista in due parti
- Cerca su una delle due parti
- **Ricerca binaria**
- Ogni volta ricerca su una parte piu' piccola
 \rightarrow prima o poi arrivera' a trovare l'elemento o a cercare nella lista vuota

Ricerca **sequenziale** e **binaria**

- In entrambi i casi: **ripetere una sequenza di istruzioni**
- **Ricerca sequenziale:** ciclo \Rightarrow ripete la sequenza sulla stessa lista con diverso stato iniziale
- **Ricerca binaria:** ripete la sequenza come sottocompito della ricerca in corso \rightarrow **ricorsione**
 - Varie **attivazioni della procedura**
 - Una attiva, le altre sospese
 - Ogni attivazione sospesa attende che un'altra termini per continuare

Sistema **ricorsivi**

- **Condizione di terminazione** (caso base)
- Istruzioni che assicurano che verra' soddisfatta prima o poi
- **Inizializzazione**, **modifica**, **verifica terminazione**
- Di solito verifica del caso base prima della ripetizione
- **Non verificata** → altra attivazione su un sottoproblema piu' vicino alla terminazione
- **Verificata** → termina l'attivazione corrente e non ne attiva altre

Nell'algoritmo di ricerca binaria...

- **Inizializzazione**: attivazione di ricerca su intera lista
- **Caso base**: trovare valore cercato o cercare su lista vuota
- **Modifica**: nuove attivazioni che cercano stesso valore in una lista piu' piccola
- Lista finita, ogni fase ricorsiva su una lista piu' piccola → prima o poi valore trovato o lista vuota → **termina sempre**

Esercizio 1 (ricerca binaria)

- Ricerca binaria di **Joe** nella lista (Alice, Bob, Carol, David, Evelyin, Fred, George, Henry, Irene, Joe, Karl, Larry, Mary, Nancy, Oliver): **quali nomi vengono esaminati?**
- Henry, Larry, Joe

Esercizio 2 (ricerca binaria)

- Ricerca binaria, ricerca di **J** nella lista A,B,C,D,E,F,G,H,I,J,K,L,M,N,O
Quali lettere vengono esaminate?
- Quali se si cerca **C**?
- Quali se si cerca **Z**?



Esercizio 3: Sequenza di Fibonacci

```
ultimo ← 0;  
corrente ← 1;  
while (corrente < 5) do  
(stampa valore di corrente;  
temp ← ultimo;  
ultimo ← corrente;  
corrente ← ultimo + temp)
```

1. Corpo del ciclo?
2. Inizializzazione?
3. Modifica?
4. Terminazione?
5. Numeri stampati?