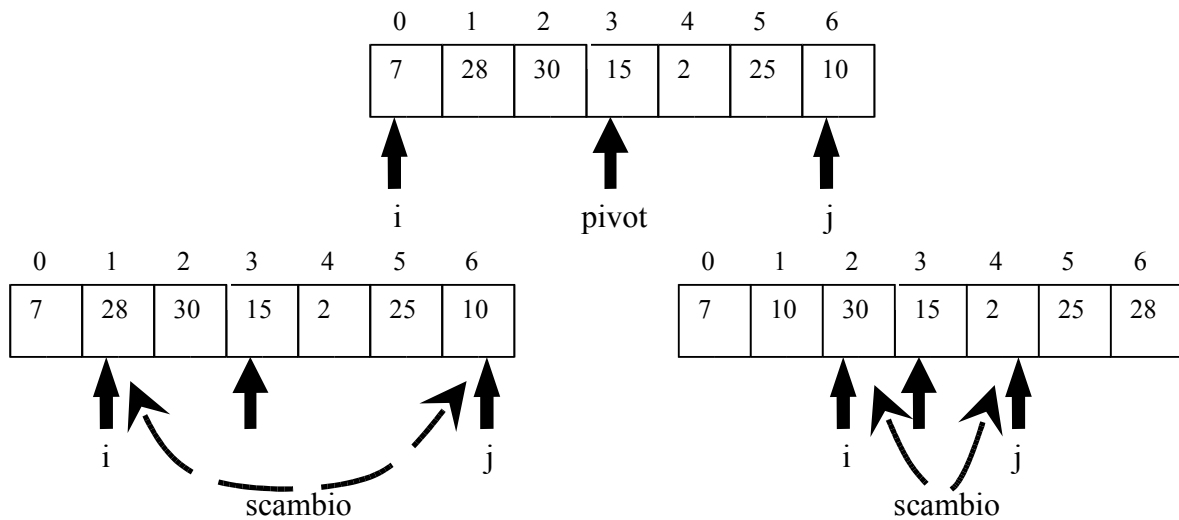


**Esercizio 1**

... continuare ricorsivamente sui 2 sottoarray: da *inizio* fino a *j* e da *i* fino a *fine*

**1.2**

```
#include <stdio.h>
#define N 20
```

```
void quickSort(int v[ ], int inizio, int fine);
```

```
main(int argc, char * argv[]) {
    int v[N], i;
    char linea[L];
    FILE *fp;

    if ( (FIN=fopen( argv[1] ,"w")) == NULL ) {
        printf("Non posso aprire il file di origine");
        return;
    }

    for(i=0; i<N; i++) {
        printf("inserire intero ");
        scanf("%d",&v[i]);
    }

    quickSort(v,0,N-1);

    for(i=0; i<N; i++) {
        fprintf(fp, "/t %d \n",v[i]);
    }

    fclose(fp);
}
```

Algoritmo di quick Sort disponibile sui lucidi

## Esercizio 2

Individuare entità e relazioni

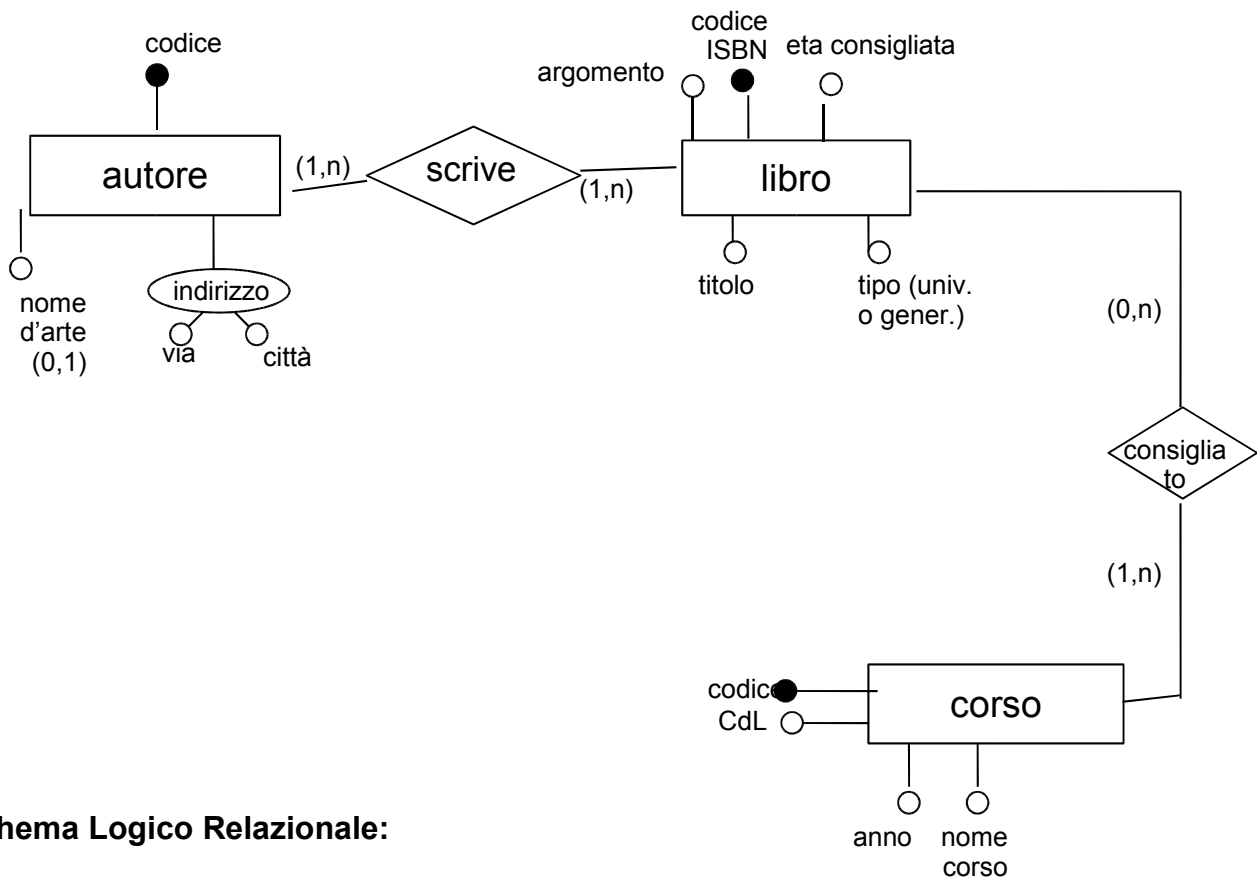
Entità: libro, testo, corso, autore

**Errore grave inserire l'entità casa editrice, perché contiene 1 sola istanza!!!**

Relazioni: gestisce, contratto, riguarda immobile.

Esiste 1 gerarchia di generalizzazione sulle entità: libro, libroDiTesto, libroGenerico

La gerarchia viene tradotta nel modello logico mediante collasso verso l'alto (scelta non vincolante, si poteva procedere anche attraverso il collasso verso il basso).



### Schema Logico Relazionale:

autore (codice, Via, città, CAP, ...)

libro (ISBN, titolo, tipo, argomento \*, etaConsigliata\*)

scrive (CodAutore, ISBN)

*CodAutore chiave forestiera di autore*

*ISBN chiave forestiera di libro*

Corso (Codice, CdL, nomeCorso, anno, ...)

consigliato (CodCorso, ISBN)

*CodCorso chiave forestiera di Corso*

*ISBN chiave forestiera di libro*

### QUERY SQL

```
Select libro.titolo FROM libro, corso, consiglia to WHERE corso.anno= 2 AND
Corso.codice=consigliato.codCorso AND consiglia to.ISBN=libro.ISBN
```