

Tabelle hash

# Tabelle hash

- Soluzione standard per il problema del **dizionario dinamico**:
  - **Insert(S,x)**: aggiungi chiave x ad S
  - **Search(S,x)**: determina se x appartiene ad S

# Tabelle hash

- Soluzione standard per il problema del **dizionario dinamico**:
  - **Insert(S,x)**: aggiungi chiave x ad S
  - **Search(S,x)**: determina se x appartiene ad S

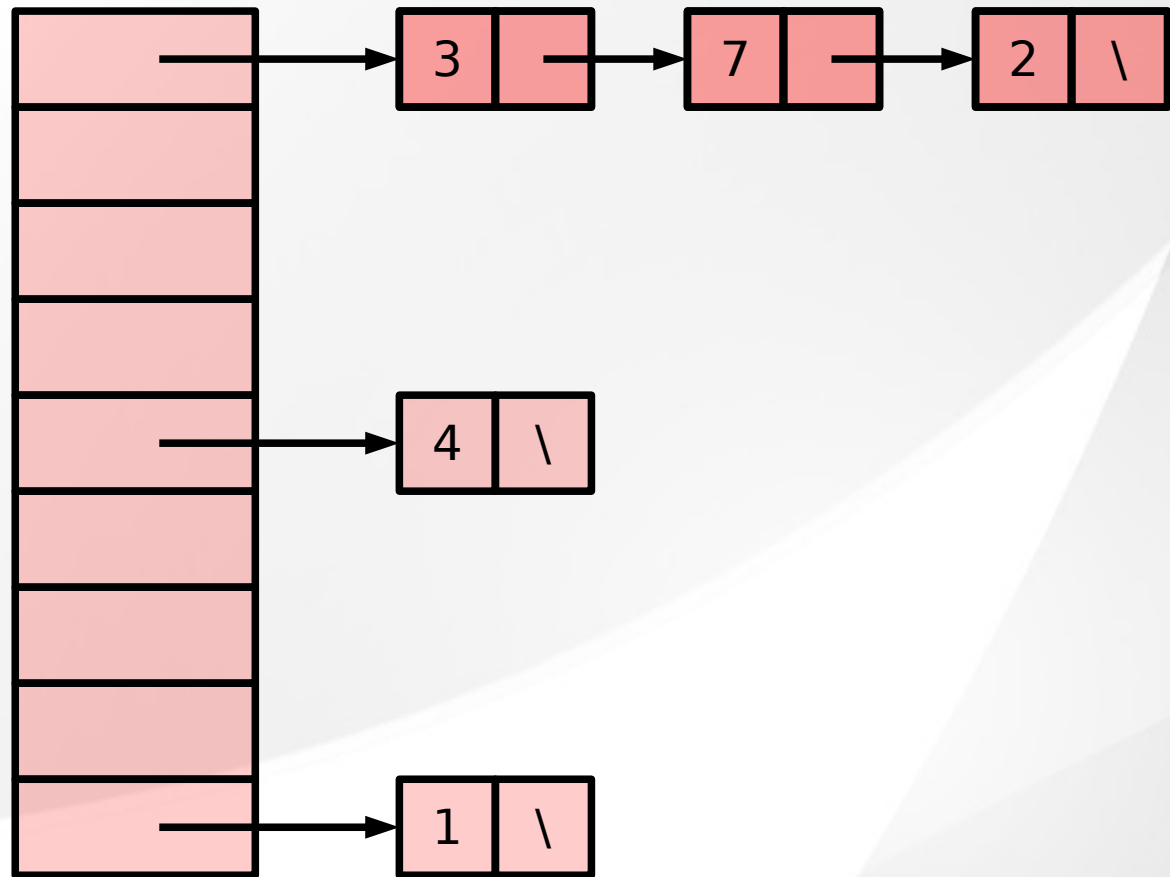
Hashing è una possibile soluzione.

Gestione dei Conflitti tramite:

- liste di trabocco
- indirizzamento aperto
- ...

# Inserzione/Ricerca

$h: U \rightarrow T$

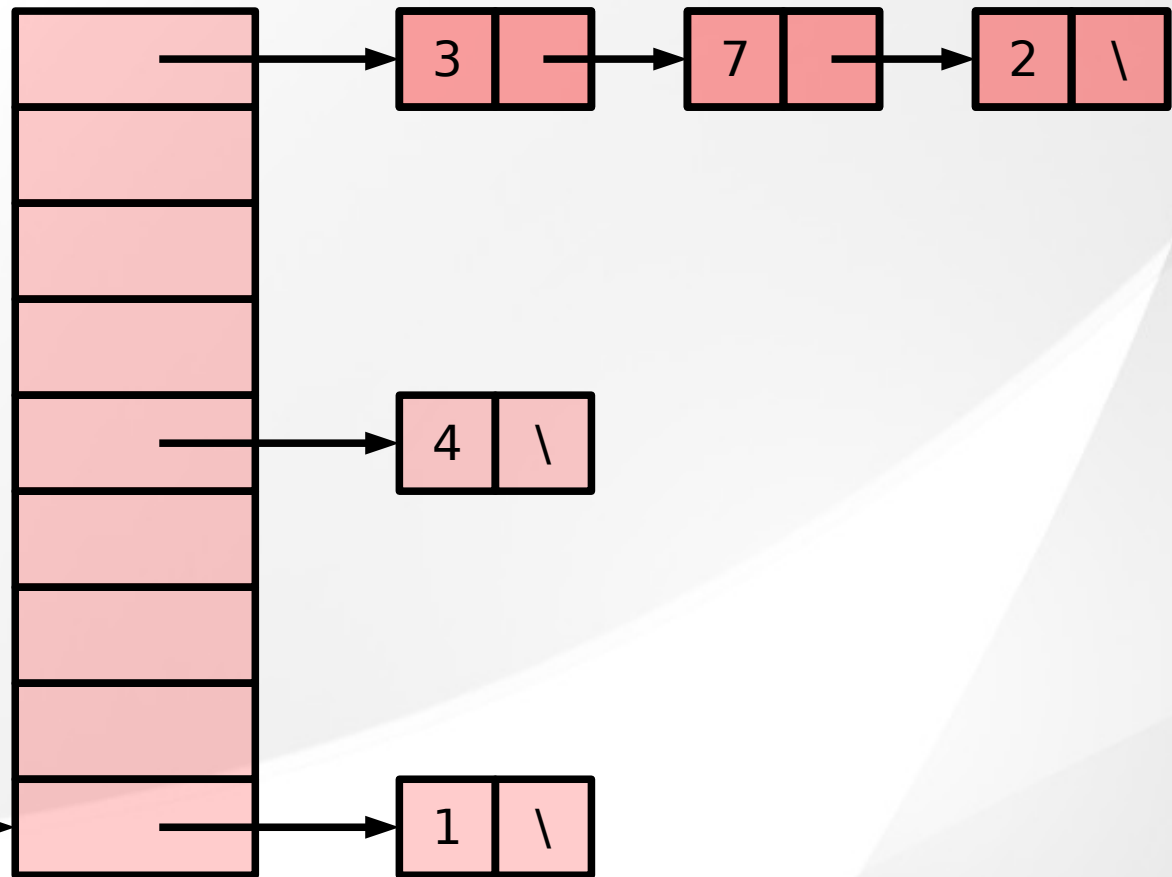


# Inserzione/Ricerca

$h: U \rightarrow T$

key: 9

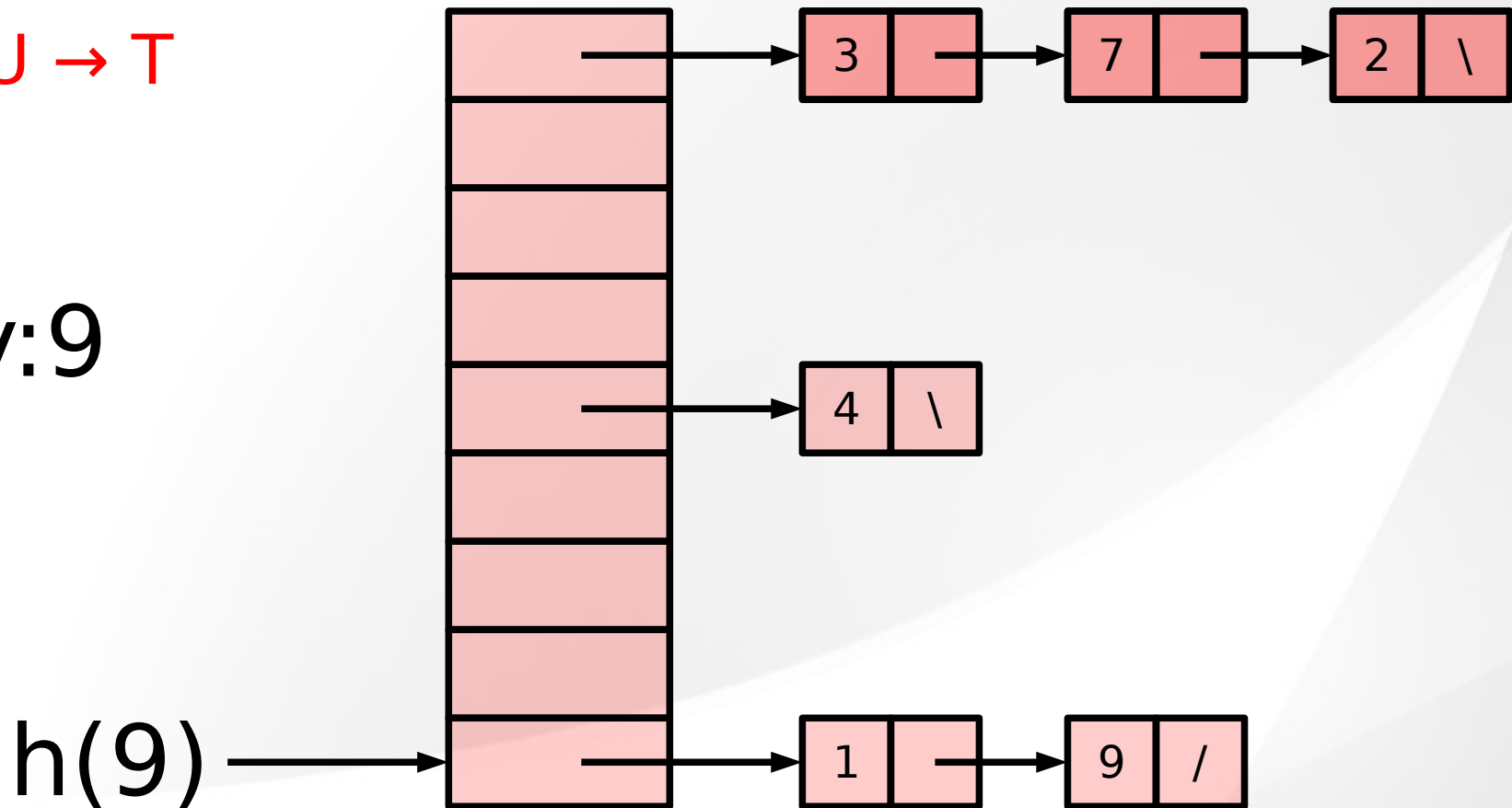
$h(9)$



# Inserzione/Ricerca

$h: U \rightarrow T$

key: 9



# Prototipo

```
typedef struct lista_el {
    int dato;
    struct lista_el *succ;
} elemento;

int main() {
    elemento ** tabella = NULL;
    int n, dimensione;
    ...
    dimensione = 2*n;
    tabella =
        (elemento **)malloc(dimensione*sizeof(elemento*));
    if(tabella == NULL) exit(1);
    for(i=0; i < dimensione; i++)
        tabella[i] = NULL; // ogni lista e'vuota
    // leggere gli elementi e inserirli nella lista corretta
    ...
}
```

# Prova Esame

- <http://ornellaia.isti.cnr.it:8888/>