

Liste

☐ Mon, 9 May

Struttura

| Lista Semplice

Una **Lista Semplice** è strutturata in questo modo:

- Esiste un *puntatore* `head` detto "*di testa*" che punta al *primo elemento* della lista.
- In ogni elemento c'è un *puntatore* `next` che punta all'*elemento successivo*.
- Ogni elemento presenta anche un `key field` "*campo chiave*" contenente i *dati*.
- Per *scorrere* una *lista semplice* si utilizza un *puntatore di sostegno* `X` che assume l'indirizzo dei vari elementi partendo da `head` e utilizzando i *puntatori* `next`.

| Lista Doppia

Una **Lista Doppia** è strutturata in questo modo:

- Ogni elemento è caratterizzato da due puntatori, `next` e `prev`. Il puntatore di `next` funziona allo stesso modo delle *liste semplici*, mentre quello di `prev` punta all'elemento precedente al corrente.
- Anche in questo caso si utilizza il puntatore di `head` per indirizzare la *prima* cella della lista, e occasionalmente si utilizza il puntatore di `tail`, "*di coda*" per indirizzare l'*ultimo* elemento della lista.

Formalmente una lista è identificata dal suo **puntatore di testa**.

Funzioni

Esistono delle **funzioni di base** il cui funzionamento deve essere garantito con le **liste a puntatori**:

- ▼ `Search(L, K)`
K è il key value che sto cercando.
- ▼ `Insert(L, X)`
X è il puntatore a tutte le informazioni che devo inserire.
- ▼ `Delete(L, X)`
Cancellare l'elemento a cui punta X
- ▼ `Min(L)`
Ritorna il minimo di S
- ▼ `Max(L)`
Ritorna il massimo di S
- ▼ `Prev(L, K)`
Ritorna il predecessore di K
- ▼ `Next(L, K)`
Ritorna il successore di K

Ulteriori operazioni che è possibile definire sono:

- ▼ `Free(X)`
Ritorna se o meno X è una cella libera
- ▼ `Alloc(sizeof(Record))`

