

Algoritmi per liste

Lista: tripla $L = (E, t, s)$

- E = insieme di elementi
- $t \in E$ = testa
- s = relazione binaria su E con queste proprietà:

- $\forall e \in E, (e, t) \notin s$
- $\forall e \in E, \text{ se } e \neq t \exists! e' \in E : (e, e') \in s$
- $\forall e \in E, \exists \text{ al più un } e' \in E : (e, e') \in s$
- $\forall e \in E, \text{ se } e \neq t \Rightarrow e \text{ è raggiungibile da } t,$
cioè

esistono $e'_1, \dots, e'_{k-1} \in E$ con $k \geq 2$: $e'_1 = t, (e'_i, e'_{i+1}) \in s$
 $\forall 1 \leq i \leq k-1, e'_{k-1} = e$

Una lista L è detta ordinata se le chiavi sono disposte in modo da soddisfare la relazione d'ordine totale.

Una lista è una struttura dati dinamica lineare, e si dice:

- Singolarmente collegata: ogni elemento contiene solo l'indirizzo del successivo.
- Doppialmente collegata: ogni elemento contiene anche l'indirizzo del precedente.

L'indirizzo dell'elemento successivo all'ultimo è indefinito, tranne nel

caso dell'implementazione circolare l'accesso ad un qualsiasi elemento avviene scorrendo tutti i precedenti a partire dalla testa.

Problemi classici:

- Visita: attraversare tutti gli elementi almeno una volta.
- Ricerca: dato un valore, stabilire se il valore è contenuto nella lista.
- Inserimento: dato una lista, inserire se possibile il valore nella lista al posto giusto.
- Rimozione: dato un valore, rimuoverlo dalla lista.

Algoritmi di visita, ricerca e inserimento

- Algoritmi di visita: se la lista ha n elementi, e l'elaborazione si svolge al più in due passi, la complessità è:

$$T(n) = 1 + n(1+d+1) + 1 = (d+2)n + 2 = O(n)$$

- Algoritmi di ricerca: restituisce l'indirizzo dell'elemento che contiene il valore, NULL se assente.
n = elementi lista.
- Caso pessimo: valore assente.

$$T(n) = 1 + n(4+1) + 1 = 2n + 2 = O(n)$$

- Caso ottimo: valore = testa.

$$T(n) = 1 + 1 = 2 = O(1)$$

- Algoritmi di inserimento: cambiano la seconda della posizione dove effettuare l'operazione.

Inserimento in lista ordinata contenente valori distinti:

- Caso pessimo: fine lista, $T(n) = 2n + 3 = O(n)$
- Caso ottimo: inizio lista, $T(n) = 4 = O(1)$
- Algoritmi di rimozione: cambia la seconda della posizione.

Rimozione da lista ordinata con valori:

distinti:

- Caso peggiore: $T(n) = 2n+7 = O(n)$

- Caso ottimo: $T(n) = 9 = O(1)$

Algoritmi di rimozione e inserimento per code

Coda: lista governata dal principio First in - First out.

Gli inserimenti e le rimozioni hanno luogo solo alle estremità.

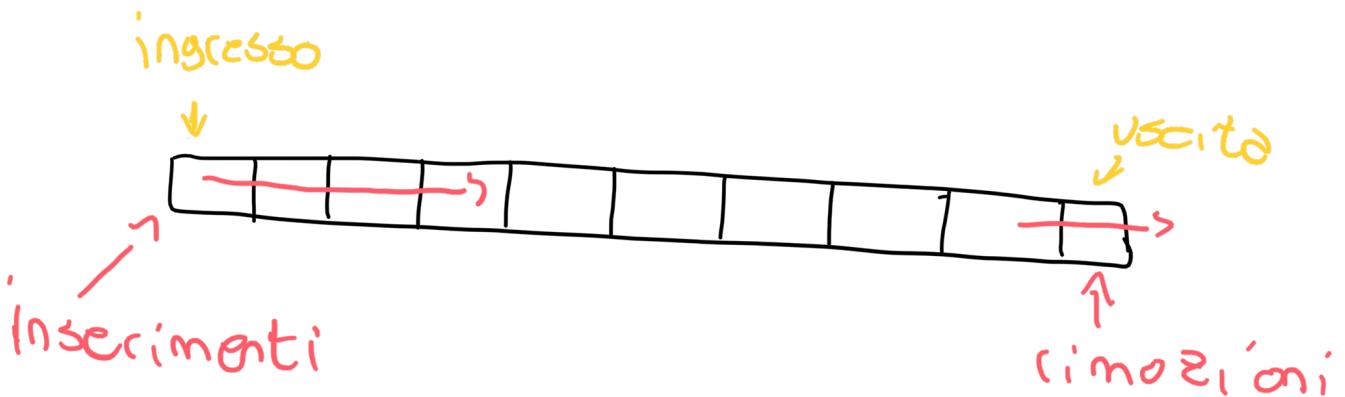
È individuata tramite l'indirizzo del primo elemento (uscita) e dell'ultimo (ingresso).

- Inserimento di valore: $T(n) = 6 = O(1)$

- Rimozione di valore: sia nel caso ottimo (coda vuota) che nel caso peggiore (coda di un elemento, $T(n) = O(1)$).

Algoritmi di rimozione e inserimento per coda

Pila: lista governata dal principio last in - first out.



Individuata tramite l'indirizzo del primo elemento, detto **cima**.

- Inserimento di valore: $T(n) = 4 = O(1)$
- Rimozione di valore: $T(n) = O(1)$ sia nel caso pessimo che ottimo.