

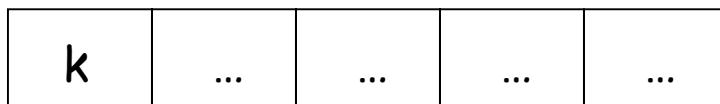
Strutture di dati

Introduciamo alcune strutture di dati che verranno, in seguito, utilizzate per creare algoritmi efficienti.

Insieme dinamico = insieme di oggetti (che nel tempo può crescere, contrarsi, etc.) i cui campi possono essere esaminati e manipolati da un algoritmo.

Oggetto = insieme di dati (a cui si fa riferimento con un unico nome) dotati di una organizzazione interna. I dati che compongono l'oggetto vengono chiamati campi.

Oggetto



chiave

dati satellite

x = puntatore all'oggetto

Operazioni su insiemi dinamici

Oggetto

| | | | | |
|---|-----|-----|-----|-----|
| k | ... | ... | ... | ... |
|---|-----|-----|-----|-----|

chiave

dati satellite

x = puntatore all'oggetto

Le operazioni su insiemi dinamici possono essere raggruppate in due categorie:

- **Interrogazioni** - restituiscono informazioni sull'insieme;
- **Operazioni di modifica** - modificano l'insieme.

Search(S,k) - dato un insieme S ed un valore chiave k restituisce un puntatore x ad un elemento in S tale che $key[x] = k$

Insert(S,x) - inserisce in S l'elemento puntato da x

Delete(S,x) - rimuove da S l'oggetto puntato da x

Dizionario = insieme dinamico che consente di effettuare le Operazioni di ricerca, inserimento, rimozione.

Esempio di insieme dinamico

Dizionario per la gestione del corso di algoritmi

| N.ordine (puntatore) | Nome (chiave) | Voto Parziale (dati satellite) |
|-------------------------|------------------|-----------------------------------|
| 1) | Panozaqi | ???? |
| 2) | Facciolo | ???? |
| 3) | Businaro | ????? |
| 4) | ... | |
| 5) | Destro | |
| 6) | ... | |
| 7) | Destro | |
| 8) | ... | |
| 9) | | |
| ... | | |

Discutiamo la rappresentazione di insiemi dinamici utilizzando semplici strutture dati che utilizzano puntatori. In particolare:

Pile;

Code;

Liste concatenate;

Alberi radicati.

Pile e code

Pile e code sono insiemi dinamici in cui l'elemento rimosso dalla operazione delete e' prestabilito:

Pile - Last-in, first-out (LIFO)

Esempio: pila di piatti,
ascensore,
.....

Code - first-in, first-out (FIFO)

Esempio: fila di persone,
.....

Pila

Realizzazione di una pila con un array. Un array di n elementi puo' essere utilizzato per realizzare un pila contenente al piu' n elementi.

1 attributo

| | | | | | | | |
|----|---|---|---|--|--|--|--|
| 15 | 6 | 2 | 9 | | | | |
|----|---|---|---|--|--|--|--|

$\text{top}[S] = 4$

| | | | | | | | |
|----|---|---|---|----|--|--|--|
| 15 | 6 | 2 | 9 | 17 | | | |
|----|---|---|---|----|--|--|--|

$\text{top}[S] = 5$

| | | | | | | | |
|----|---|---|---|----|--|--|--|
| 15 | 6 | 2 | 9 | 17 | | | |
|----|---|---|---|----|--|--|--|

$\text{top}[S] = 4$

Stack-Empty(S)

If ($\text{top}[S]=0$)

 then return true

 else return false

Push(S, x)

$\text{top}[S] \leftarrow \text{top}[S] + 1$

$S[\text{top}[S]] \leftarrow x$

Ognuna delle 3 operazioni
richiede un tempo $O(1)$

Pop(S)

if Stack-Empty(S)

 then error "underflow"

 else $\text{top}[S] \leftarrow \text{top}[S] - 1$

 return $S[\text{top}[S]+1]$

Code

Realizzazione di una coda con un array. Un array di n elementi puo' essere utilizzato per realizzare un coda contenente al piu' $n-1$ elementi.

2 attributi

| | | | | | | | |
|--|--|--|----|---|---|---|--|
| | | | 15 | 6 | 2 | 9 | |
|--|--|--|----|---|---|---|--|

head[Q] = 4. tail[Q] = 8

| | | | | | | | |
|---|--|--|----|---|---|---|----|
| 3 | | | 15 | 6 | 2 | 9 | 17 |
|---|--|--|----|---|---|---|----|

head[Q] = 4. tail[Q] = 2

| | | | | | | | |
|---|--|--|----|---|---|---|----|
| 3 | | | 15 | 6 | 2 | 9 | 17 |
|---|--|--|----|---|---|---|----|

head[Q] = 5. tail[Q] = 2

Enqueue(Q,x)

$Q[\text{tail}[Q]] \leftarrow x$

if ($\text{tail}[Q] = \text{length}[Q]$)

then $\text{tail}[Q] \leftarrow 1$

else $\text{tail}[Q] \leftarrow \text{tail}[Q] + 1$

Le 2 operazioni

richiedono un tempo $O(1)$

Dequeue(Q)

$x \leftarrow Q[\text{head}[Q]]$

If $\text{head}[Q] = \text{length}[Q]$

then $\text{head}[Q] \leftarrow 1$

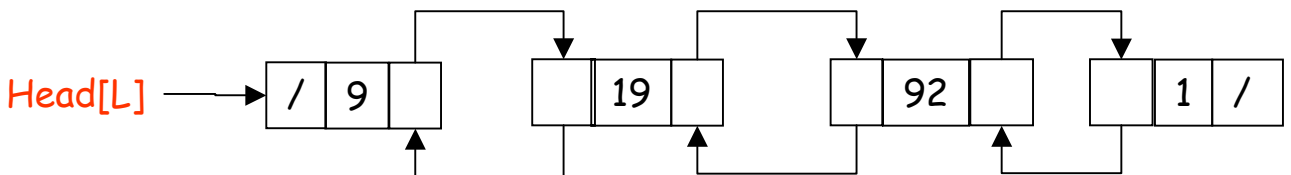
else $\text{head}[Q] \leftarrow \text{head}[Q] + 1$

Return x

Liste concatenate

Una lista concatenata e' una struttura di dati in cui gli oggetti sono sistemati secondo un ordine lineare. L'ordine e' determinato attraverso puntatori.

Lista concatenata doppia (bidirezionale) = insieme di oggetti composti da tre campi - una chiave, un campo puntatore next ed un campo puntatore prev.



Ricerca in una lista concatenata

Trova il primo elemento con chiave k nella lista concatenata L

List-Search(L, k)

$x \leftarrow \text{head}[L]$

While ($x \neq \text{NIL}$) and ($\text{key}[x] \neq k$)

do $x \leftarrow \text{next}[x]$

Return x

$$T(n) = \Theta(n)$$

Inserimento in una lista concatenata

Inserisce un oggetto x all'inizio della lista concatenata L

List-Insert(L, x)

$\text{Next}[x] \leftarrow \text{head}[L]$

If $\text{head}[L] \neq \text{NIL}$

then $\text{prev}[\text{head}[L]] \leftarrow x$

$\text{Head}[L] \leftarrow x$

$\text{Prev}[x] \leftarrow \text{NIL}$

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

Cancellazione da una lista concatenata

Cancella un oggetto x dalla lista concatenata L

List-Delete(L, x)

If $\text{prev}[x] \neq \text{NIL}$

then $\text{next}[\text{prev}[x]] \leftarrow \text{next}[x]$

else $\text{head}[L] \leftarrow \text{next}[x]$

If $\text{next}[x] \neq \text{NIL}$

then $\text{prev}[\text{next}[x]] \leftarrow \text{prev}[x]$

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