

Strutture Dati

corso regionale di preparazione per le OII

Giorgio Audrito

Università degli Studi di Torino

30 Settembre 2010

Cosa si intende con struttura dati?

Una *struttura dati* è due cose insieme:

- ▶ Un insieme di variabili per contenere i dati.
- ▶ Un certo numero di funzioni per accedere, inserire, modificare questi dati: le **uniche** che possono essere usate per elaborare i dati contenuti.

Strutture basate su *array*

Pila

La struttura dati *Pila* serve per contenere oggetti come se fossero “impilati”: cioè l'unico elemento a cui si può accedere è l'ultimo inserito.

Interfaccia

- ▶ `void push(t x)`: aggiunge in cima alla pila l'elemento `x`.
- ▶ `t peek()`: restituisce il valore dell'elemento `x` attualmente in cima alla pila.
- ▶ `t pop()`: restituisce il valore dell'elemento `x` attualmente in cima alla pila e lo rimuove dalla pila stessa.

Come la si può implementare?

Strutture basate su *array*

Coda

La struttura dati *Coda* serve per contenere oggetti come se fossero “in coda”: cioè l'unico elemento a cui si può accedere è il primo inserito.

Interfaccia

- ▶ `void push(t x)`: aggiunge in fondo alla coda l'elemento `x`.
- ▶ `t peek()`: restituisce il valore dell'elemento `x` attualmente in cima alla coda.
- ▶ `t pop()`: restituisce il valore dell'elemento `x` attualmente in cima alla coda e lo rimuove dalla coda stessa.

Come la si può implementare?

Strutture basate su *array*

Osservazioni

- ▶ Entrambe le strutture dati appena viste sono implementate tramite array.
- ▶ Entrambe le strutture dati appena viste sono pensate per aggiungere elementi senza limiti.
- ▶ Sarebbe quindi necessario basarle su una struttura dati più generica *array dinamico* in grado di ridimensionarsi durante il processo.

Array Dinamico

Un array che consente cancellazione e inserimento arbitrari.

Interfaccia

- ▶ `int push(t x)`: aggiunge l'elemento `x` in fondo all'array, eventualmente ridimensionandolo, e restituisce l'indice in cui l'ha inserito.
- ▶ `t pop()`: rimuove l'elemento in fondo all'array, eventualmente ridimensionandolo.
- ▶ `t operator[] (int i)`: restituisce l'elemento `V[i]`.
- ▶ `t assign(int i, t x)`: esegue `V[i]=x`, dove `i` deve essere un indice già presente nell'array.