

Contenedores

Biblioteca Estándar de Templates

Agustín J. González

ELO320

Contenido

Contenedores:

Vector, List, Deque, Stack, Queue, Priority Queue, Set,
Map

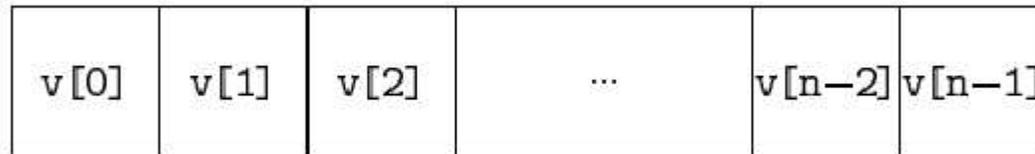
Iteradores

Tipos para organizar colecciones (Contenedores)

Structure	Addition of new element	Removal of first element	Removal of middle element	Inclusion test
Vector indexed, random access to elements, bounded size	$O(1)$ or $O(n)$ †	$O(n)$	$O(1)$ or $O(n)$ †	$O(n)$ or $O(\log n)$ ‡
List sequential access to elements, rapid insertion and removal	$O(1)$	$O(1)$	$O(1)$	$O(n)$
Deque random access, rapid insertion to front and back	$O(1)$	$O(1)$	$O(n)$	$O(n)$ or $O(\log n)$ ‡
Stack insertion and removal only from front	$O(1)$	$O(1)$	NA	NA
Queue insertion only from front, removal only from back	$O(1)$	$O(1)$	NA	NA
Priority Queue rapid removal of largest element	$O(\log n)$	$O(1)$ or $O(\log n)$ §	NA	NA
Set ordered collection, unique values, rapid insertion, removal and test a multiset allows repeated elements	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$
Map collection of key-value pairs a multimap allows multiple elements with same key	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$
Notes	†constant if accessing existing position, linear if inserting/removing ‡logarithmic if ordered, linear if not ordered §constant access time, logarithmic removal			

Vectores

- Son una generalización de un arreglo
- Provee eficiente acceso aleatorio a elementos.
- Operaciones de alto nivel como incremento o reducción dinámica del tamaño.



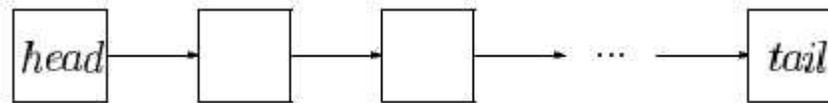
- Acceso a un elemento toma tiempo $O(1)$

Strings

- En un sentido un string es un arreglo de caracteres
- En otro sentido es un tipo de dato de bajo nivel enteramente diferente por su manipulación y tipo de operaciones.
- Hay muchas operaciones específicas para strings
- `string aName = "Paula González";`

Listas (List)

- Su tamaño es arbitrario, su implementación hace un uso eficiente de la memoria al crecer o reducir su tamaño.
- Solo acceso secuencial.
- Eficiente inserción y remoción de cualquier posición.



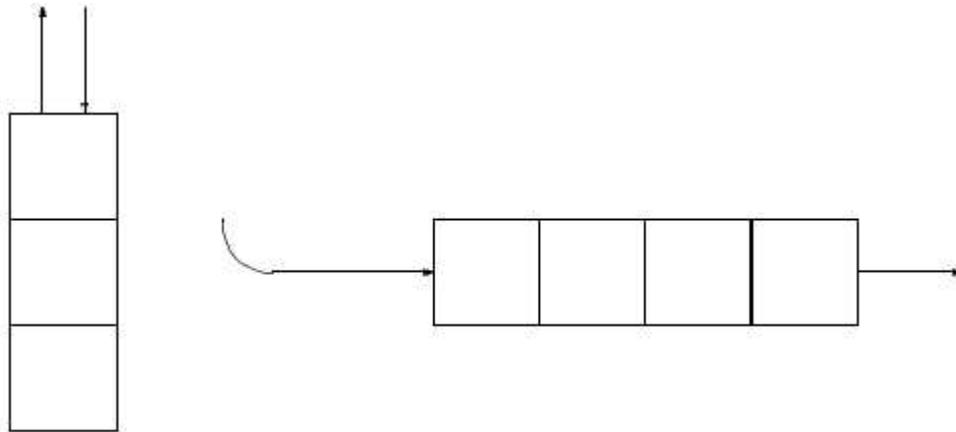
Deque (Double Ended Queue)

- Crece y se encoge según sea necesario
- Eficiente inserción y remoción de cada extremo
- Acceso aleatorio a elementos



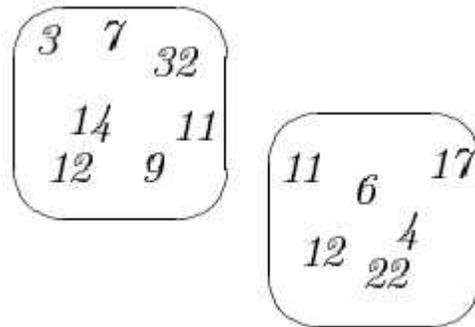
Stacks y Queues

- Especializadas formas de colas doblemente enlazadas (deque)
- Stack es LIFO (Last in, First Out)
- Queue es FIFO (First in First Out)



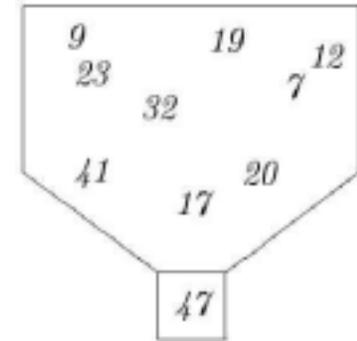
Conjuntos: Sets

- Colección no ordenada de elementos
- Inserción, remoción y test de pertenencia en tiempo logarítmico
- Permite operaciones como Unión, diferencia.
- Existen también los Multiset que permiten mantener múltiples copias de cada elemento.



Colas de Prioridad

- Eficiente inserción de valores nuevos (logarítmica)
- Eficiente acceso al valor mas grande (o menor) (logarítmica)
- Remoción logarítmica de elementos.



Asociaciones o mapeos: Map

- Colección pares de claves y valores
- Las claves pueden ser cualquier tipo de datos ordenable (los elementos se pueden comparar. Ej. String)
- Los valores pueden ser cualquier tipo
- Eficiente inserción, remoción, prueba de inclusión

$key_1 \rightarrow value_1$
 $key_2 \rightarrow value_2$
 $key_3 \rightarrow value_3$
...
 $key_n \rightarrow value_n$

Criterios para seleccionar un contenedor

- Como serán accedidos los elementos?
 - Random => vector o deque
 - ordenada => conjuntos o map
 - secuencial => lista
- Es importante el orden en el cual los elementos son mantenidos?
 - Ordenada => set
 - puede ser ordenada => vector o cola
 - depende de orden de inserción => stack o queue
- Cambiara el tamaño de la colección ampliamente durante la ejecución?
 - Si => use list o set
 - No => use un vector o deque
- Es posible estimar el tamaño de la colección?
 - Si => use un vector
- Es la colección indexada? (la colección puede ser vista como una serie de pares claves/valor)
 - índices son valores enteros => use vector o deque
 - De otra manera use un map

Criterios para seleccionar un contenedor (cont.)

- Puede ser relacionados los valores?
 - Set requieren operadores relacionales
 - Vector o list no lo requieren
- Búsqueda y remoción del mas grande una operación frecuente?
 - Si => use colas de prioridad (mas rápida que set)
- En que posiciones los valores son insertados o removidos de la colección?
 - Inserciones en el medio de listas son eficientes
 - Inserciones en el medio de un vector no lo son
 - Stack y queue solo pueden insertar en extremos
- Es frecuente la operación de mezcla de dos o mas secuencias en una?
 - Si datos son ordenados => use set
 - De otra manera => use una list

Iteradores (iterators)

- Problema básico: ¿Cómo se accede a los elementos de una colección sin conocer como la colección esta organizada?
- Solución: la colección define un nuevo tipo de dato creado especialmente para lasos de repetición (for, while, etc)
- Un gran numero de algoritmos ya están implementados por la biblioteca estándar (STL). Todos ellos usan iteradores.

¿Cómo se describe un rango de valores?

- Normalmente se describe usando el valor de partida y el valor posterior al ultimo.
- El valor posterior al ultimo no es parte de la colección, solo una marca.

Operaciones con Iteradores

- **Begin /end:** Por convención, los contenedores retornan el valor de partida (una referencia en realidad) en respuesta a la función `begin()`, y retornan el valor posterior al ultimo con `end()`.
- Un iterador puede ser comparado por igualdad con otro iterador. Son iguales cuando apuntan a la misma posición.
- Un iterador se puede des-referenciar usando el operador `*`. De esta forma se accede al valor denotado por el iterador.
- Un iterador puede ser incrementado, así se referencia al próximo elemento en secuencia. Se usa operador `++`.