

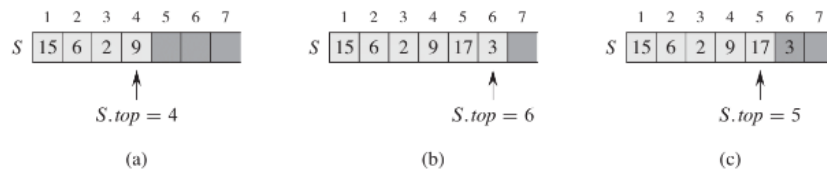
Estructura de datos elemental.

Pilas y cola son opciones dinámicas son conjuntos dinámicos en los que el elemento eliminado del conjunto por la operación de eliminación se especifica previamente

En un apilamiento, el elemento eliminado del conjunto es el que se insertó más recientemente: la pila implementa la política de abandono, salida o LIFO.

PILAS.

La operación INSERT en una pila a menudo se llama PUSH, y la operación DELETE, que no toma un argumento de elemento, a menudo se llama POP. Estos nombres son alusiones a apilamientos físicos, como los montones de platos que se usan en las cafeterías. El orden en que las placas se sacan de la pila es el inverso del orden en el que se insertaron en la pila, ya que solo la placa superior es accesible.



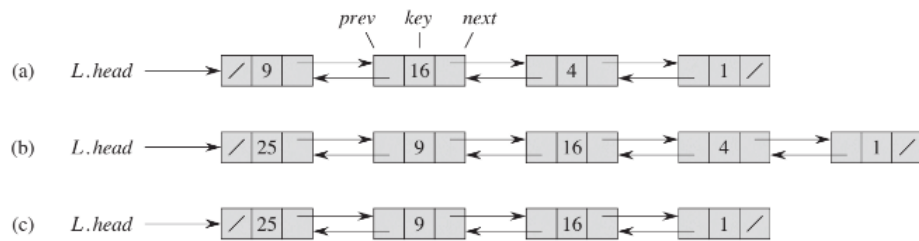
COLAS.

La propiedad FIFO de una cola hace que funcione como una línea de clientes esperando pagarle a un cajero. Cuando un elemento está en cola, toma su lugar al final de la cola, del mismo modo que un cliente recién llegado toma un lugar al final de la línea. El elemento eliminado siempre es el que está a la cabeza de la cola, como el cliente al frente de la línea que ha esperado el mayor tiempo

LISTAS ENLAZADAS.

Una lista enlazada es una estructura de datos en la que los objetos se organizan en un orden lineal. Sin embargo, a diferencia de una matriz en la que el orden lineal está determinado por los elementos de la matriz, el orden en una lista vinculada está determinado por un puntero en cada objeto. Las listas vinculadas proporcionan una representación simple y flexible para conjuntos dinámicos, que admiten (aunque no necesariamente de manera eficiente).

Si una lista está vinculada, omite el puntero previo en cada elemento. Si la lista no está ordenada, los elementos pueden aparecer en cualquier orden. En la lista dinámica, el puntero previo del encabezado de la lista apunta a la cola, y el siguiente puntero de la cola de la lista apunta a la cabeza.



REPRESENTANDO ÁRBOLES ENRAIZADOS.

Podemos extender el esquema para representar un árbol binario a cualquier clase de árboles en los que el número de hijos de cada nodo sea a lo sumo una constante k : reemplazamos los atributos arrojados por el extremo por niño1; niño2; :::; niño k . Este esquema ya no funciona cuando el número de hijos de un nodo no está limitado, ya que no sabemos cuántos atributos (matrices en la representación de múltiples matrices) asignar en avance.

Además, incluso si el número de niños está limitado por una gran constante, pero la mayoría de los nodos tienen un número pequeño de niños, podemos perder mucha memoria.

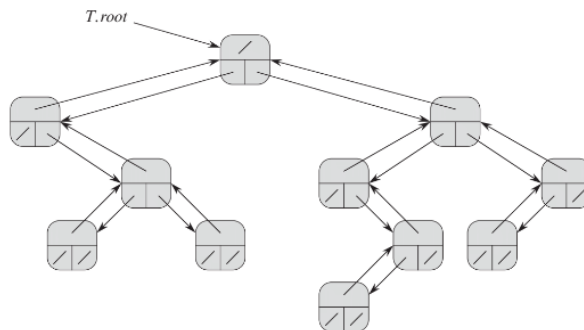


Figure 10.9 The representation of a binary tree T . Each node x has the attributes $x.p$ (top), $x.left$ (lower left), and $x.right$ (lower right). The key attributes are not shown.

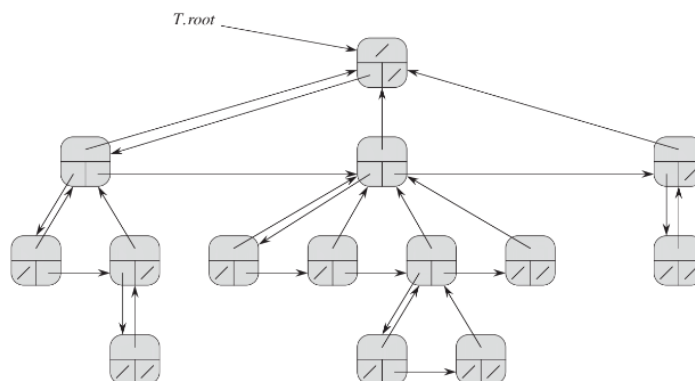


Figure 10.10 The left-child, right-sibling representation of a tree T . Each node x has attributes $x.p$