

Pila (stack)

Lista lineal donde los elementos se insertan eliminan siempre por el mismo extremo, es una estructura LIFO.

Usaremos esta estructura cuando no tengamos que quitar o insertar elementos del principio o de enmedio.

La representación gráfica de una pila es como la de cualquier estructura dinámica:

top

NIL

Siempre añadiremos o eliminaremos elementos por el Top. Para ello disponemos de dos funciones: Push, para agregar un elemento y Pop, para eliminarlo.

Algoritmo POP (var top1:apuntador);

Var p:apuntador;

Begin

If Top = Nil then pila vacía

else

begin

p:=top;

top:=top^.siguiente;

dispose(p); *

end;

end;

* Se podría guardar una variable con el dato, antes de eliminar ese elemento de la lista.

* En algunos lenguajes a veces no hace falta ponerlo. Nosotros sí lo pondremos para liberar ese espacio.

Algoritmo Push (var top1: apuntador, dato:...);

var p :apuntador;

Begin

```

New (p);

p^.sig:=top;

p^.info:=dato;

top:=p;

end;

```

En estos algoritmos de tratamiento de pilas representadas por apuntadores deberíamos tener en cuenta los casos en que ya no dispongamos de espacio.

Cola (queue)

Lista lineal donde las inserciones siempre se hacen por un extremo y las eliminaciones siempre por el otro. Es, por tanto, una estructura FIFO.

La representación gráfica es como la de la pila:

top bottom

Nil

Dispondremos de un apuntador más.

Cuando la cola esté vacía llegaremos a la siguiente situación:

Nil Nil

Algoritmo INSERTAR(dato, var top1: ^nodo, var bottom: ^info);

```

Begin

New(p);

p^.info:=dato;

if bottom=Nil then

begin

top:=p;

bottom:=p;

end

else

begin

```

bottom^.sig:=p;

bottom:=p;

end;

p^.sig:=nil;

end;

Algoritmo Eliminar(var top:^.nodo);

Begin

if top=Nil then (cola vacía)

else

begin

p:=top;

top:=top^.sig;

dispose(p);

end;

end;

Como ejercicio, se propone el determinar una estructura de datos óptima para gestionar una partida de dominó.

También, dada la lista siguiente, se pregunta cómo se puede eliminar un nodo de una manera eficiente:

X X X