4. Implementaciones de pilas basadas en referencias

Carlos Zerón Martínez

Universidad Nacional Autónoma de México

zeron@ciencias.unam.mx

Jueves 5 de Noviembre de 2020

Implementación de una pila basada en una lista ligada

Para no utilizar la clase nodo para hacer referencia al último nodo agregado a la pila, se emplea directamente una lista ligada con referencia al nodo inicial como estructura de datos auxiliar.

La cabeza de la lista será por convención el tope de la pila, dado que es la única referencia que necesitamos distinguir.

La excepción del método push desaparece porque no hay limitaciones de espacio en esta implementación.

Implementación de una pila basada en una lista ligada

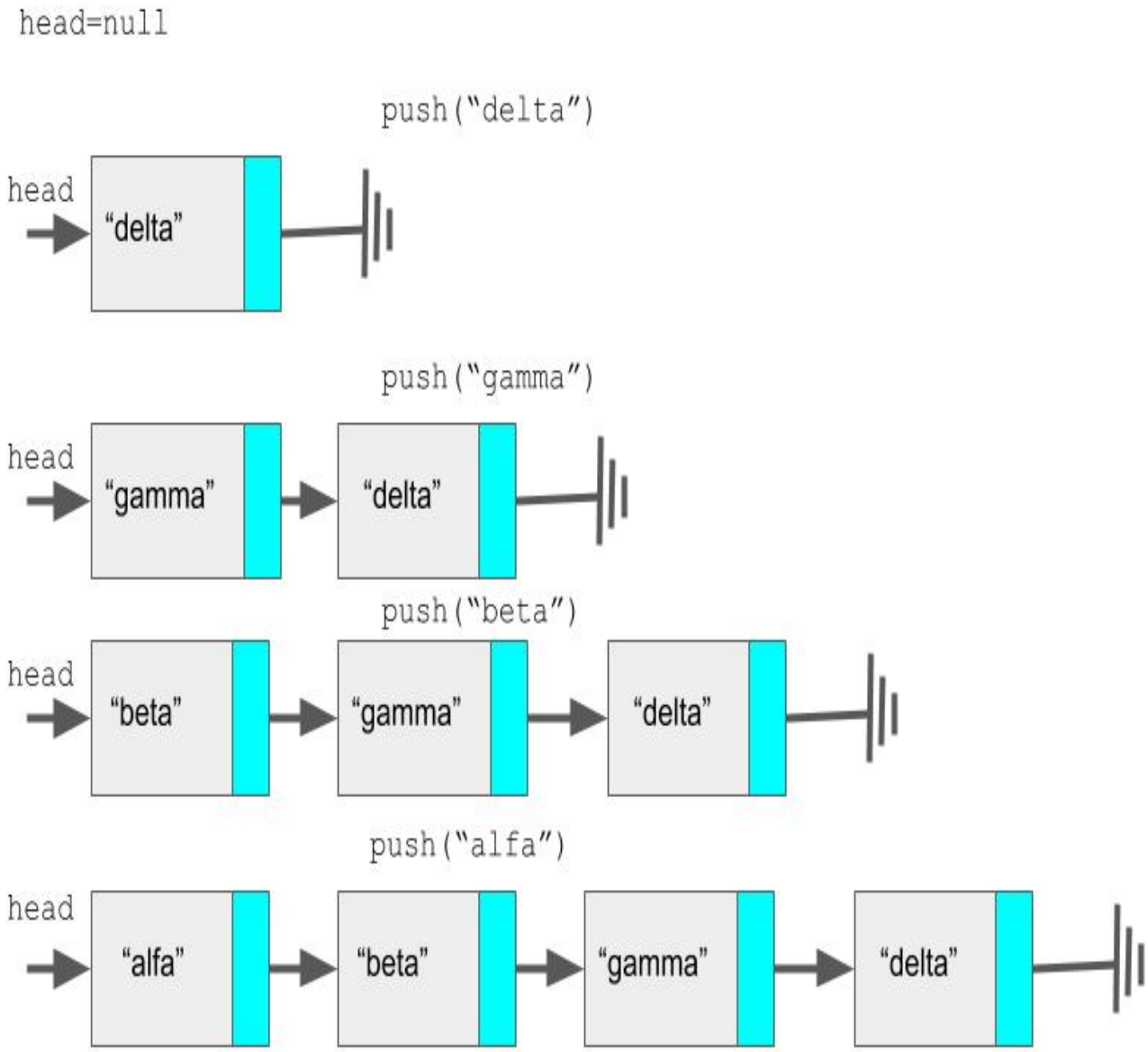
```
public class PilaLista implements Pila {

    private final ListaLigada lista;

    /**
     * Constructor de una pila
     * basada en una lista ligada.
     */
    public PilaLista() {
        lista = new ListaLigada();
    }

    @Override
    public boolean isEmpty() {
        return lista.isEmpty();
    }

    @Override
    public int size() {
        return lista.size();
    }
```

Implementación del método push

Simplemente se inserta en la posición que representa el tope (la primera posición de la lista ligada)

```
@Override
public void push(Object obj) {
    lista.add(0, obj);
}
```

Secuencia de inserciones

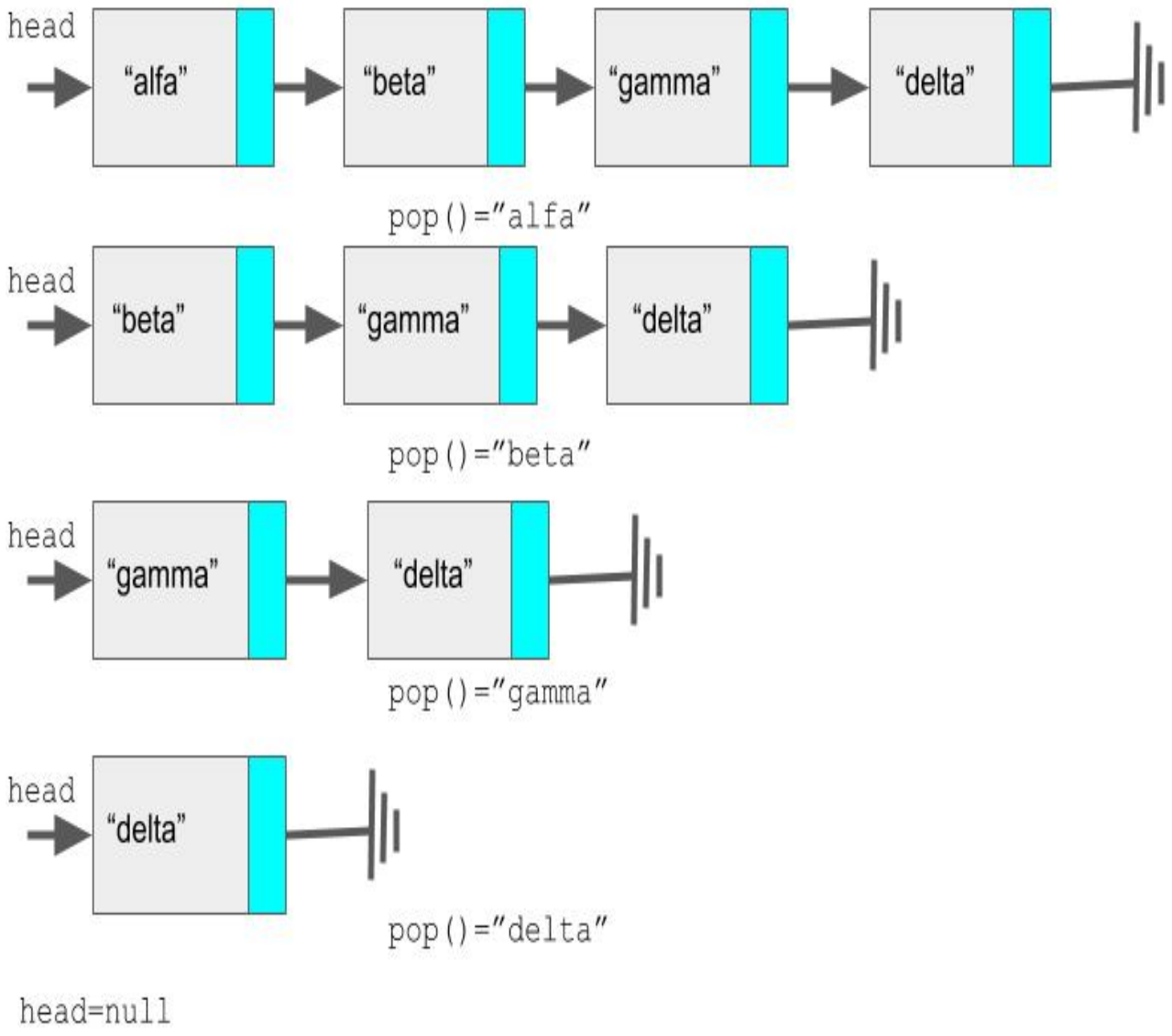


Implementación del método pop

Si hay elementos, se rescata el elemento del tope (la primera posición de la lista ligada), se elimina de ahí mismo y se devuelve el elemento rescatado.

```
@Override
public Object pop() {
    if (lista.isEmpty()) {
        return null;
    }
    Object tmp = lista.get(0);
    lista.remove(0);
    return tmp;
}
```

Secuencia de eliminaciones



Implementación del método peek

Si hay elementos en la pila, regresa el que está en el tope (primera posición de la lista ligada).

```
    @Override
    public Object peek() {
        return lista.isEmpty() ? null : lista.get(0);
    }
}
```

Análisis de complejidad

Tener la implementación de listas ligadas ayuda porque las operaciones de pilas pueden definirse en términos de las operaciones de listas, aplicándolas en el caso particular de las listas ligadas.

Como el tope de la lista está en la primer posición, el tiempo de ejecución es $O(1)$ en todos los métodos

Implementación de pilas con una lista doblemente ligada

Se puede escoger cualquiera de los dos extremos por convención como el tope (la primera o la última posición), pero hay que ser consistentes en todas las operaciones.

Como en el caso de la implementación con una lista ligada simple, podemos tener como atributo una lista doblemente ligada y todas las operaciones podemos definirlas también en términos de listas para el caso de referencias dobles

La complejidad se mantiene constante para todas las operaciones, sin embargo, a menos de que por alguna razón se requiera el acceso secuencial a más posiciones que el tope, no es la implementación a elegir, debido a que se emplea más memoria, precisamente porque se conservan dos referencias por cada nodo, cuando con la lista ligada se tiene una sola.