

## 1. Primera actividad

Una estructura de datos para indexar claves numéricas tiene las siguientes características:

Existen dos tipos de nodos, el nodo *no terminal* que sirve para indicar cual de los hijos contiene el nodo que hace referencia al siguiente dígito de la clave. Por otro lado, está el nodo terminal. Este nodo es una nodo hoja que contiene la dirección de memoria en disco donde se encuentra la información del registro con esa clave.

Por ejemplo, en la figura 2 se muestra la estructura al ingresar la clave 121.

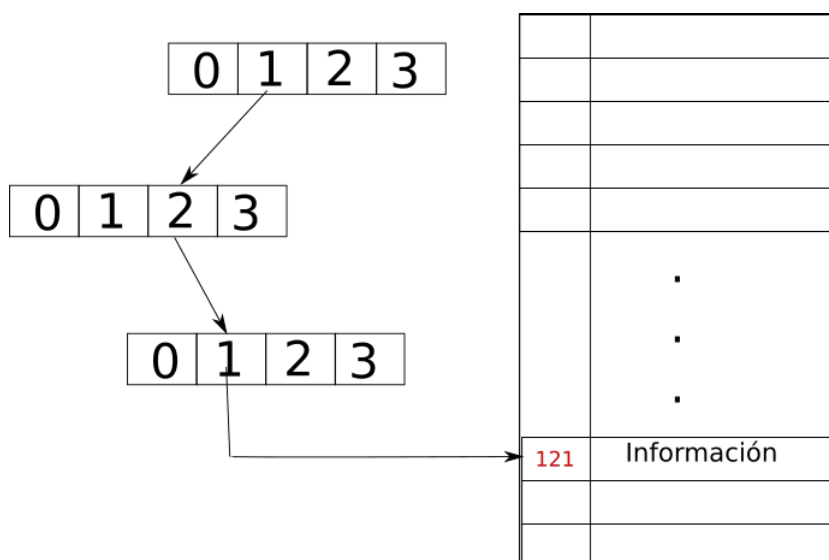


Figura 1: Resultado de la estructura al ingresar la clave 121

Luego, si ingresamos la clave 31, la estructura de datos toma la siguiente forma:

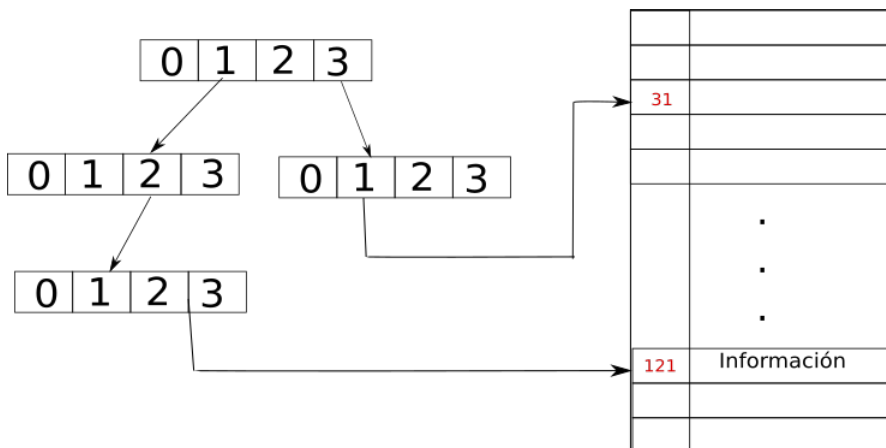


Figura 2: Resultado de la estructura al ingresar la clave 31

Se pide implementar esta estructura de datos, de modo que pueda indexar el archivo que se le proporcionará para la etapa de test.

Las operaciones que se piden son:

1. Insertar una clave
2. Retornar la información en disco de una clave  $K$
3. Retornar la información en disco de todas las claves que inician con el dígito  $ab$ , por ejemplo si ingresa 13 deberá retornar los registros de todas las claves que inician con 13 (131, 136, 137, 1345...)

## 2. Segunda actividad

Para un *Ternary Search Tree* y *Radix Tree* se pide indexar un archivo de texto y realizar las siguientes operaciones:

1. Mostrar todas las palabras que inician con un sufijo ingresado por teclado.
2. Indicar las línea y columna en el texto donde se encuentra una palabra. Tenga en cuenta que la palabra puede repetirse varias veces en el texto y por lo tanto deberá retornar una lista de líneas columnas.
3. Calcular la cantidad de memoria RAM que ocupan ambas estructuras e indicar cual consume menos RAM.
4. Implementar eliminación en ambas estructuras.
5. Indicar cuantas veces se repite una palabra en el texto.
6. Tome 100 palabras aleatorias en el texto para realizar consultas en las estructuras y compute el número promedio de nodos que son visitados hasta llegar al nodo terminal.