



Universidad Andrés Bello
Facultad de Ingeniería
Ingeniería en Computación e Informática

Estructura de Datos Proyecto Árbol 4-ario

Integrantes: Esteban Sepúlveda Perez
Victor Jofre Miranda
Claudio Vásquez Campos
Profesor: Tomas Lara
Fecha de Entrega: 21-06-2017

Enunciado

Se pretende crear una nueva estructura, la que llamaremos NE. La cual representará una imagen digital en blanco y negro, donde una imagen puede ser representada en un espacio rectangular compuesto de $N \times N$ píxeles (en potencia de a dos), cada uno representando un punto (blanco o negro) de la imagen. La representación usual de una imagen es una matriz de $N \times N$, con 0 los píxeles blancos y con 1 los negros, como se ve en la Figura 1.

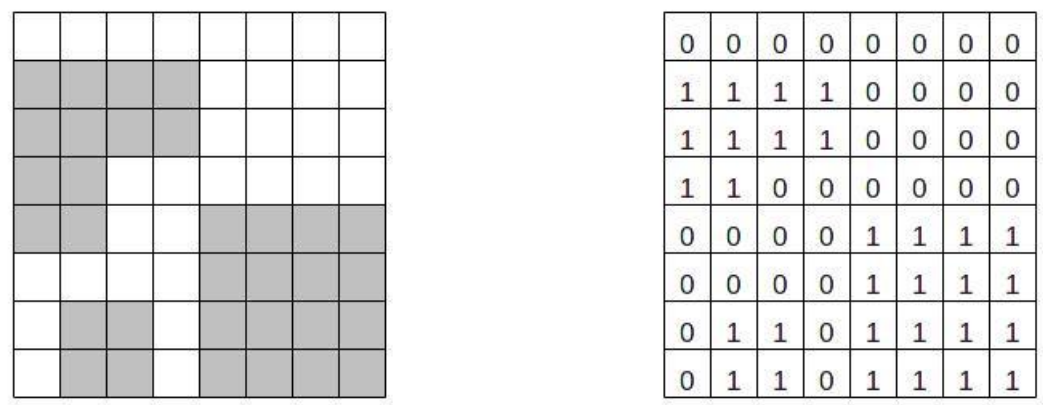


Figura 1: Representación de una imagen en matriz

NE es un árbol 4-ario que permite representar de manera compacta una imagen, que en otro caso podrá ocupar grandes cantidades de memoria. NE tiene tres tipos de elementos: blancos, negros y grises, que representan un grupo de píxeles de la imagen según el esquema recursivo de representación planteado en la Figura 2, en a) blancos, b) negros y c) grises.

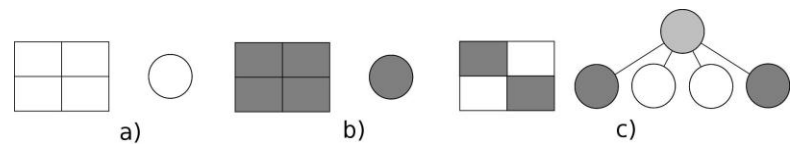


Figura 2: Esquema recursivo de representación

La raíz del NE representa la imagen completa: si la imagen es toda negra, el NE tiene un solo elemento con valor NEGRO. Si la imagen es completamente blanca, tiene un solo elemento con valor BLANCO. En cualquier otro caso, la imagen se representa con un nodo GRIS, y se le asocian 4 subárboles, cada uno representando uno de los cuadrantes resultantes al dividir la imagen en 4 cuadros iguales. Al continuar este proceso recursivamente se obtiene la representación completa de la imagen por medio de una estructura arborescente compacta.

A modo de ejemplo, para la imagen que aparece en la Figura 1, la raíz del NE que la representa debe ser un nodo gris, para indicar que no todos los píxeles de la imagen son del mismo color. Luego, se divide la imagen en 4 cuadrantes y se aplica recursivamente el mismo proceso, asociando NE resultantes como subárboles, como se aprecia en la Figura 3.

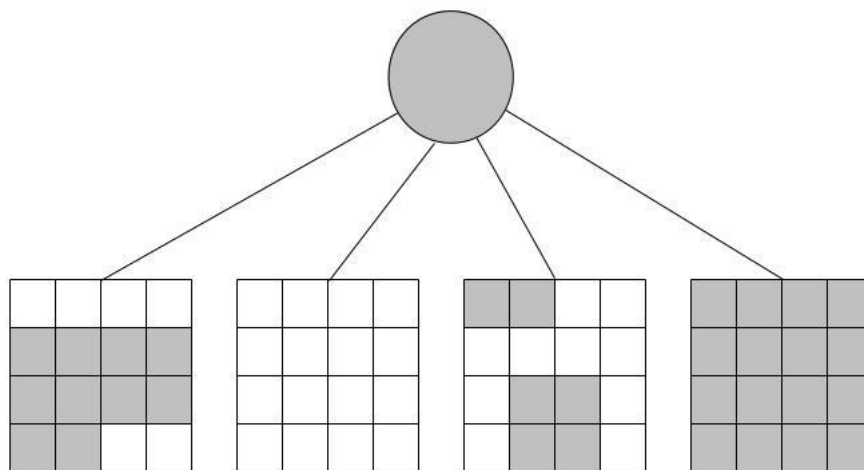


Figura 3: Esquema recursivo de representacion

Este proceso sigue así, hasta llegar al siguiente arbol 4-ario, que representa la imagen completa, el cual se visualiza en la Figura 4.

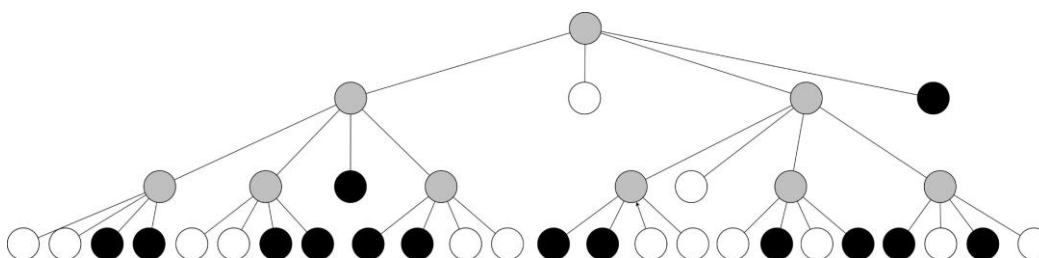


Figura 4: Conversión de la matriz a la estructura NE

Se solicita crear un programa, en el lenguaje C, que permita visualizar la imagen representada en el archivo datos.dat. Considere una matriz de 100 100. El archivo tiene códigos 0 (BLANCO), 1 (NEGRO), y 2 (GRIS), y el recorrido es por niveles.

Consideraciones

1.- **Datos.dat** : Este archivo contiene en cada LINEA el nodo padre o raíz y proseguido de este vienen sus hijos u hojas.

Consideramos lo siguiente, si el valor es:

- .- Si es 2 el padre contendrá 4 hijos y puede contener hijos con nodos 2, 1 y 0.
- .- Si es 1 el padre es completamente negro, y no se subdivide en ningún hijo.
- .- Si es 0 el padre es completamente blanco, y no se subdivide en ningún hijo.

```
1 2
2 2
3 0
4 2
5 1
6 2
7 2
8 1
9 2
10 2
11 0
12 2
13 2
14 0
15 0
16 1
17 1
18 0
19 0
20 1
21 1
22 1
23 1
24 0
25 0
26 1
27 1
28 0
29 0
30 0
31 1
32 0
33 1
34 1
35 0
36 1
37 0
38
```

2.- Del archivo datos.dat se hace una transformación a una matriz de pixeles que solo contendrá valores 1 (negro) y 0 (blanco).

```
La representacion numerica de la matriz es:
0 0 0 0 0 0 0 0
1 1 1 1 0 0 0 0
1 1 1 1 0 0 0 0
1 1 0 0 0 0 0 0
1 1 0 0 1 1 1 1
0 1 0 0 1 1 1 1
0 1 1 0 1 1 1 1
0 1 1 0 1 1 1 1
```

3.- Una vez creada esta matriz de pixeles se procede a pintar cada valor, definidos en el punto anterior (punto 2). Considerando nuestro fondo de consola negro, para valor 0 sera de color blanco y para valor 1 sera de color negro.

```
La imagen de la matriz es:
```



4.- Cuadrantes: Como en este caso el padre corresponde a un 2 se hace una división en 4 nodos Ubicamos los nodos en los siguientes cuadrantes, con el siguiente orden.

1	3
2	4

5.- Funcion:

La funcion encontrar pixel busca dentro de la matriz de pixeles la coordenada ingresada (columna, fila), y entrega el sector donde se ubicara (cuadrante) este funcionamiento esta indicado en el punto anterior.

```
Ahora determinaremos en que cuadrante se encuentra un pixel
Ingrese la columna del pixel: 8
Ingrese la fila del pixel: 8
La posicion del pixel con la coordenadas <8,8>, esta en el cuadrante: 4
```

6.- Conversion del Archivo a Matriz NE:

En esta funcion se reciben los valores del archivo datos.dat y se ingresan en una matriz con el proposito de mostrar el arbol generado.

```
Ahora transformaremos el arbol a un arreglo
El Arreglo contiene los siguientes elementos:
2      2      0      2      1      2      2      1      2      2
0      2      2      0      0      1      1      0      0      1
1      1      1      0      0      1      1      0      0      0
1      0      1      1      0      1      0      0      0      0
```

Decisiones Implementacion:

Para ver los niveles de grafo ingresado en datos.dat decidimos ocupar busqueda en anchura, porque necesitamos revisar cada nivel del grafo formado, una parte del codigo se implemento con este tipo, y ademas se mencion que se uso fuerza bruta y no recursividad, es decir, los valores para formar el arbol se recorrieron por linea.

Las coordenadas del cuadrante se hicieron asi ya que fue mas facil para nosotros de implementar.

Se crearon bibliotecas correspondientes a traves de include llamadas "archivo.h" y "funciones.h". Archivos.h contiene datos.dat que crea y obtiene este mencionado archivo. La biblioteca funciones.h contiene las estructuras principales del arbol, nodo y cola. Tambien esta llena la matriz, determina el cuadrante, crea el arbol, transforma el arbol e imprime.