

Informe

Para dar con la solución al proyecto 3 de Ayed , se implemento un árbol de 4 hijos (quadtree) con lo que se representa cada hijo para un subcuadrante de la imagen en cuestión.

Para cada cuadrante de la imagen se le asigna un hijo que estará determinado por una parte de la matriz , la asignación se hace en el sentido de las agujas del reloj ,es decir, para el primer cuadrante se le asigna el hijo1, para el 2 el hijo 2, siempre y cuando la imagen no sea homogénea .

Ya conocida la estructura a utilizar se listaran las operaciones a realizar:

- **CrearArbol** : acción que dada una matriz con la información de la imagen los índices y los límites , verifica que la matriz sea homogénea , si lo es , se creara un nodo de ese color solamente , sino se llamara recursivamente con la subdivisión de la imagen con el cuadrante correspondiente donde tendrá el hijo 1, si es cuadrante 1 , y así sucesivamente. Cabe destacar que dependiendo de la opción del usuario se llamara a un crear arbol diferente , para el caso de la matriz se hara lo citado anteriormente pero si es la imagen es cargada en formato arbol su creación es análogo , solo que para verificar, el color lo hace mediante el string que se ingreso.
- **Cambiar colores**: acción que dado árbol no vacío, cambiar los colores de sus nodos hojas, esto se hace mediante un recorrido preorden donde se verifica primeramente si el nodo es padre y además es de color 1 o 0 , es decir , es homogénea . sino se verifica cada hijo que no sea null y se le intercambia los colores.
- **Espejo** : acción que dado un árbol no vacío intercambia los hijos de un nodo padre , en sentido que cambia el hijo 1 por el hijo 2 , y el hijo 3 , por el hijo 4.
- **Numero de colores**: acción que dado un árbol no vacío cuenta el numero de nodos negros, blancos y grises.
- **Altura**: función que dado un árbol no vacío retorna la altura del mismo.
- **Recorrido Preorden**: acción que dado un árbol no vacío hace el recorrido preorden del mismo.
- **Salvar imagen** : acción que dado un árbol no vacío , guarda el ancho, alto y el recorrido en preorden en un archivo de extensión .qt
- **Mostrar Matriz** : acción que dado un árbol no vacío , muestra la imagen en formato texto por pantalla mediante el llenado de una matriz dependiendo de como se subdivide el quadtree con sus respectivos índices.
- **Eliminar árbol** : acción que dado un árbol no vacío , hace un recorrido en posorden para ir eliminando sus hojas y luego su raíz.

Desde el punto de complejidad en espacio, el árbol si no es homogéneo y para cada subárbol a su vez esta subdividido y así sucesivamente para n nodos , se tiene que , se utilizaría unos nodos de mas que serian los padres de cada hijo , entonces llegando al caso recursivo donde el pixel sea 1 para una matriz nxn, tenemos que para poder representar todas esas subdivisiones tendríamos que usa un nodo adicional para cada subcuadrante del árbol , en ese caso , seria óptimo utilizar una matriz nxn y no un quadtree en memoria.