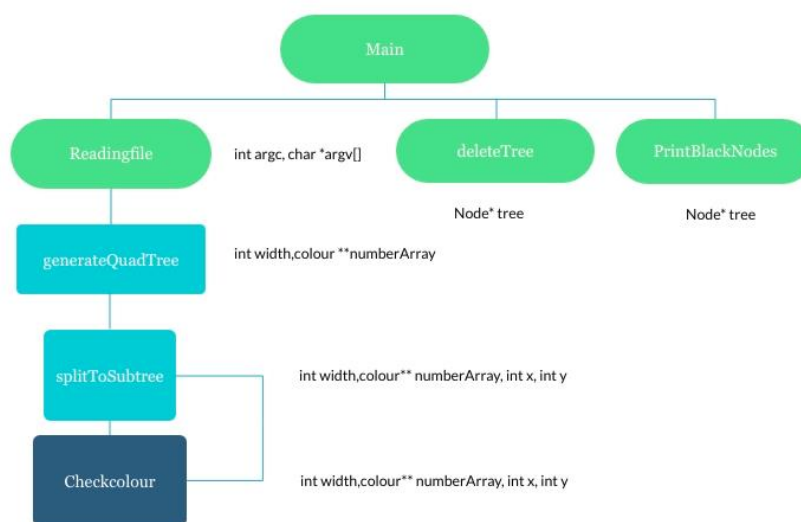


Quadtree Encoding of a Binary Image Project:

The program is designed to create a quadtree from a binary image which is inputted in the form of coordinates in a text file. The input file contains the width of the image, the number of black nodes and the coordinates of the black pixels. The program will encode a quadtree which is a type of data structure where each original or parent node has four lower-level or subsequent child nodes, where each element is continually split into four pieces [1]. From this information the program will output the location of the black nodes and their size. The details of this process is described below.

Program hierarchy chart

Indicating the inter-relationship of functions



Functions:

The main function:

Checks if the program has been called appropriately with a text files. Then, passes the information onto the Readingfile function. It also runs the PrintBlackNodes function after the other functions are run and the Deletetree Function.

The Readingfile function: Reads the file, checks for errors and stores the information in a dynamic array. For example, If the array is empty it will output an error statement. The function creates a dynamic two-dimensional array where it stores the data and uses pointers the access information. Initially, I used a simple static array implementation however the code output the wrong results at the printing stage. This may be due to how I passed the array through many functions, whereas in the current implementation the array is stored in the structure as well and can be accessed with pointers.

The GenerateQuadTree function: This function passes the array onto the splitToSubtree function with 0,0 initial x and y values. This allows the splitToSubtree function to run recursively with the correct initial input coordinates.

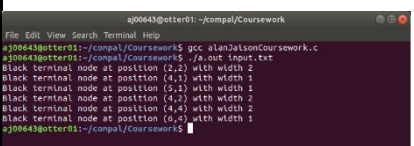
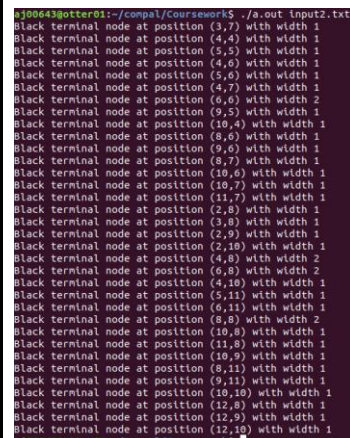
The splitToSubtree Function: The function is a recursive function that calls itself until a black or white node is found. Initially, the function calls the checkColour function which outputs true or false based on if all the values in the selected array are the same. If they are all the same, then the function checks if that point is 1 or 0 then assigns colour as white or black. This essentially creates a node at the point stating its value.

If this statement is false, then the function will call itself again but with new parameters of the location changed and the width further halved. This caused the algorithm to go further down the quadtree until black and white nodes are found in each different section of the image. At each recursion event the width and the coordinate values of x,y are assigned to the quadtree structure.

PrintBlacknodes function: When the splitToSubtree function finishes running the full quadtree for the binary image is created. The main function calls the PrintBlacknodes function which is a recursive function that iterates down the quadtree outputting each black node. A print statement will output the black nodes location and size. This information is accessed using pointers from the quadtree structure.

deleteTree function: Lastly, the deleteTree function is run which recursively it iterates through the quadtree deleting the nodes and freeing the memory. This was not necessary for the program to work but I wanted to test if it improves the time complexity.

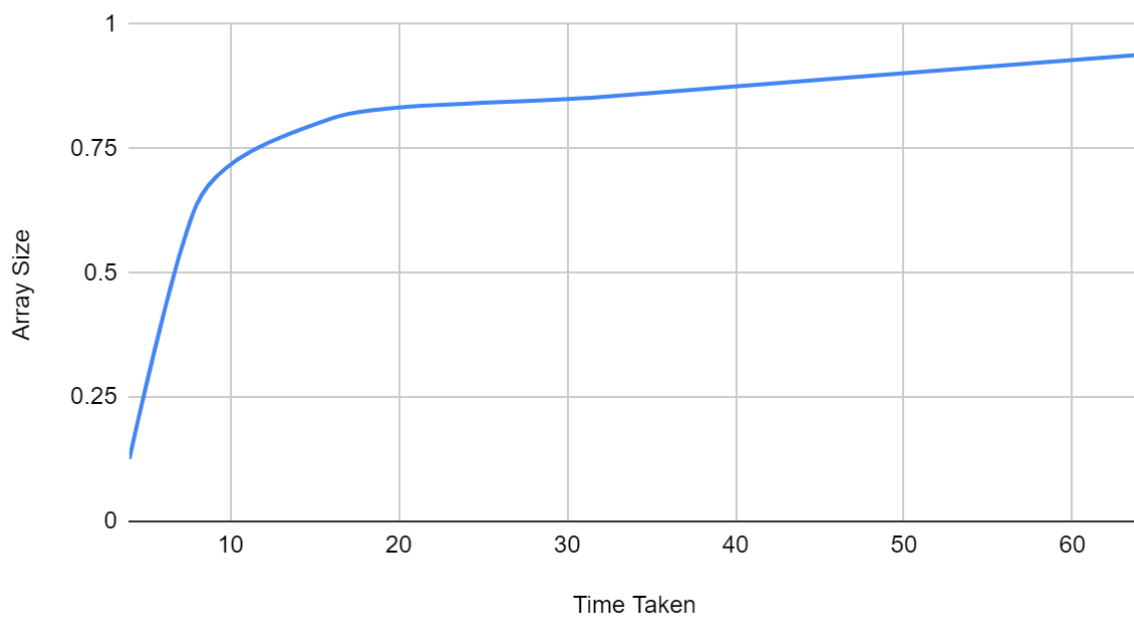
Testing:

Input:	Output:
Text.c	 <p>Obtained the expected output for the first input, Suggesting the program is correct.</p>
Text2.c	 <p>Obtained the expected output for the first input, Suggesting the program is correct.</p>

Time Complexity Test: (5,000 runs)

Array Size	Number of Black Nodes	Run Time 1	Run Time 2	Run Time 3	Run Time Average
4	2	0.1362	0.1256	0.1149	0.1255666667
8	4	0.6436	0.6253	0.6443	0.6377333333
16	8	0.8231	0.7998	0.8089	0.8106
32	16	0.8658	0.8435	0.8519	0.8537333333
64	32	0.9421	0.9287	0.9451	0.9386333333

Chart showing the time taken against array size



The time complexity for a quadtree is $O(\log(n))$ which is the expected output. The result I obtained indicates this complexity.

References:

Number	Used for:	Source link:
1	Definition of QuadTree	https://www.techopedia.com/definition/14714/quadtree
2	General code query's	https://stackoverflow.com/
3	Time complexity	SurreyLearn.com