Data Structures

Assignment 3: (Due on Nov 10 at 6 pm)

Goal: The goal of this assignment is to familiarize you with the list data structure. The particular image processing task that is chosen is galaxy boundary detection and representation.

What to code?

Your task can be divided into three main components. The first task is to clean up the images that we have been getting so far. Before we can process the Hubble images further, we need to add a simple routine to "clean-up" the thresholded image to remove single pixel isolated "galaxies" and "thin hairs" that are attached to the main blobs representing the galaxies. This can be achieved by a simple image processing operation called morphological *opening*. So, your first task is to add this morphological *opening* operation that is defined by the following sequence of two operations (erosion followed by dilation). *Let us assume that the galaxy pixels are marked with 0 and the background is 255.*

$$O(i,j) = \begin{cases} 0 & \text{if } \textbf{ALL} \text{ 3 by 3 neighbors of I(i, j) } == 0 \\ 255 & \text{otherwise} \leq N \end{cases}$$

$$O(i,j) = \begin{cases} 0 & \text{if } \textbf{ANY} \text{ 3 by 3 neighbors of I(i, j) } == 0 \\ 255 & \text{otherwise} \leq N \end{cases}$$

The morphological opening operation should be done on the thresholded image *before* depth-first or breadth-first labeling.

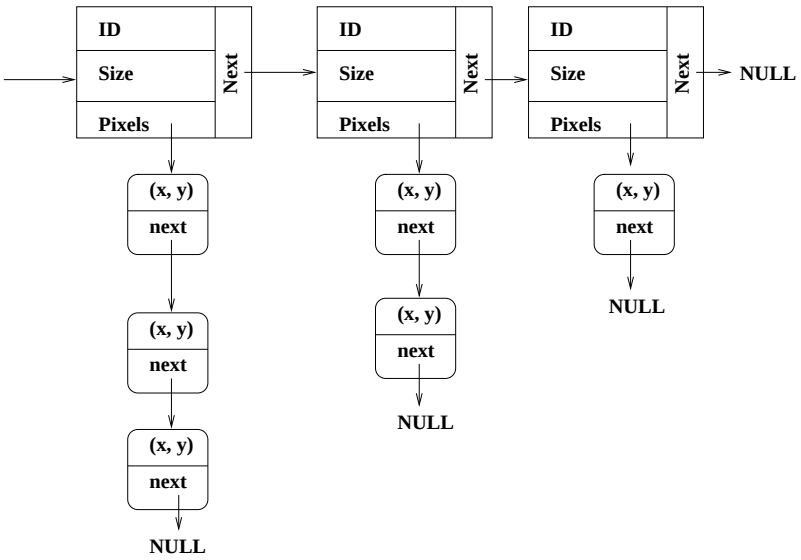
The second step would be to identify and flag the image pixels that are part of the "boundary". One easy test for identifying boundary pixels would be to check to see if the pixel has value not equal to 255 *and* has some of its boundary neighbors that are 255 and some that are not 255.

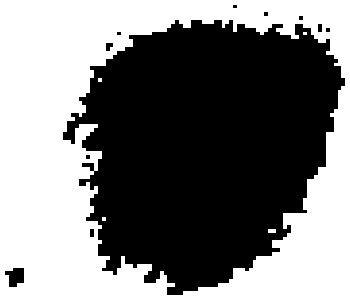
The third step would be to "chain" the boundary into a linked list. The order of the pixels in the linked list should correspond to the order along the boundary.

Note that each galaxy detected in the image would result in one boundary linked list. Store all these boundary lists in a list of lists structure where the longest boundary is the first element. Identify each boundary list by the identifier assigned to the corresponding galaxy in the depth-first and breadth-first search.

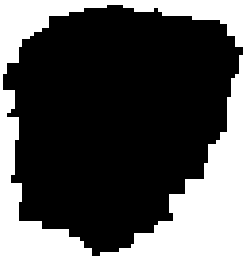
Include an option to save the 5 largest boundary chains found in the image. While writing out the boundary pixels back into an image, label them in order they appear in the corresponding list. Wrap around to a label of 0 if the number of pixels exceed 200 in a particular list.

What to submit? For details about what and how to submit, see first assignment.

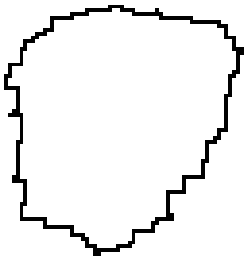




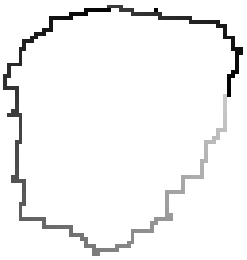
Thresholded image



Morphologically opened



Detected boundary pixels



Labeled boundary pixels

1. The morphological operations specified above assumes that your galaxies are labeled with 0 and the background is white (255) after thresholding. For vice-versa, the morphological operation would be:

$$O(i,j) = \begin{cases} 255 & \text{if } \textbf{ALL } 3 \text{ by } 3 \text{ neighbors of I(i, j) == } 255 \\ 0 & \text{otherwise} \leq N \end{cases}$$

$$O(i,j) = \begin{cases} 255 & \text{if } \textbf{ANY } 3 \text{ by } 3 \text{ neighbors of I(i, j) == } 255 \\ 0 & \text{otherwise} \leq N \end{cases}$$

   Note that one can generalize the 3 by 3 neighborhood to a M by M neighborhood to turn it into an opening over a larger neighborhood.

2. You can detect boundary easily by first `Eroding` the image and then subtracting it from the original image.
3. You can chain the edge pixels by depth first searching *but* by considering only a 4-connected neighborhood around each pixel instead of the usual 8 neighbors.
4. You should be able to instantiate the generic linked *list.ads* in your textbook to implement the list of lists structure. If you decide to implement the list of lists data structure as one big unit that is also okay.
5. You can structure the code to have the following specification files

```
image_data_type.ads    process_boundary.ads   stack.ads
list.ads               process_image.ads
list_of_list.ads       queue.ads
```