



# BLOB DETECTION

Day-3, Slot-2

---

# Outline

What is Blob ?

Why Blob ?

How to find Blobs ?

Let's code it!!

Summary



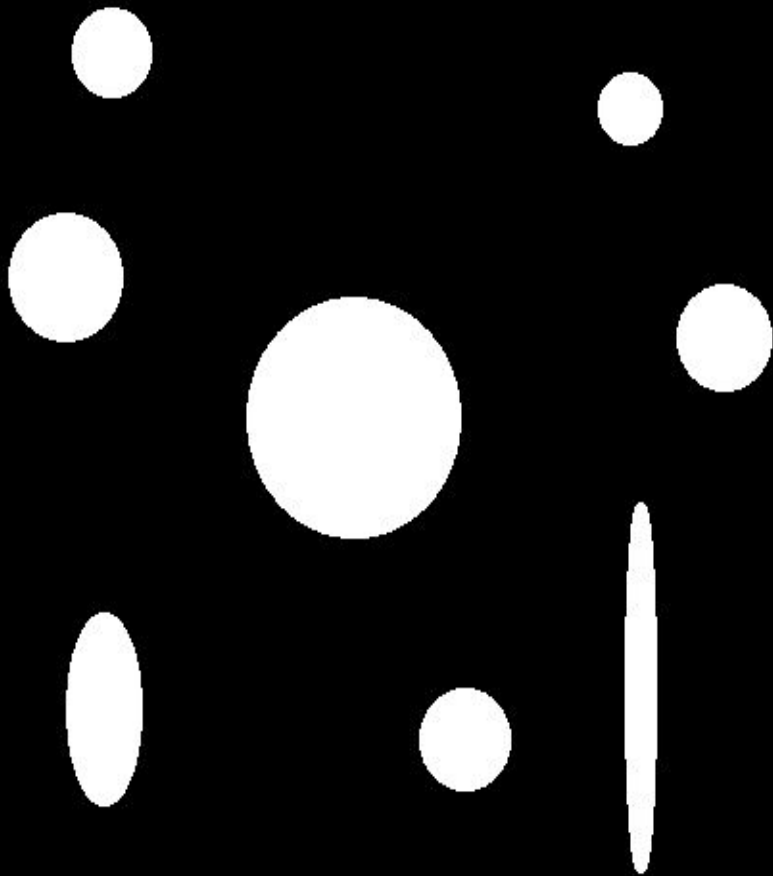
# Meaning of Blob

**Blob stands for Binary Large Object and refers to the connected pixel in the binary image.**

Blob is basically a set of adjacent pixels of the same color. This set of points is basically a region in the image.

---

**Let's find Blobs in the next slide....**



## Think about the following questions:

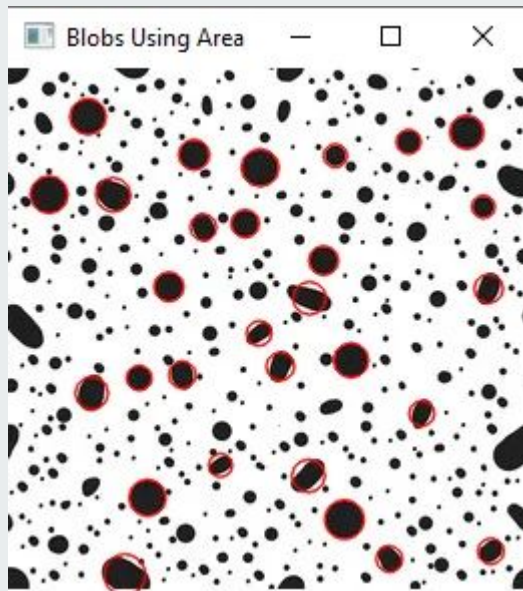
1. Is this image binary?  
Yes!!!
2. Are there any blobs in this image?  
Yes!!!
3. Number of blobs in the image?  
9
4. What common pixel value they have?  
255 for 8 blobs  
0 for 1 blob



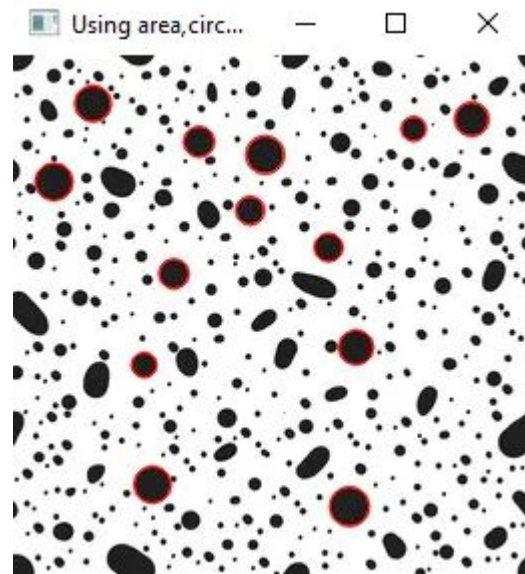
# Why we are finding blobs ?

Finding blobs is the basic step in many image processing functions.


1. Blobs check whether two regions with same pixel values are connected.
2. Sets of points returned by blobs detection function can be used to find the following values related to the region:
  - a. Area
  - b. Circularity
  - c. Convexity



Small objects rejection using area



Non circular objects rejection  
using circularity



# How to find Blobs ?

Blobs are connected pixels with same values.

Procedure to find blobs:

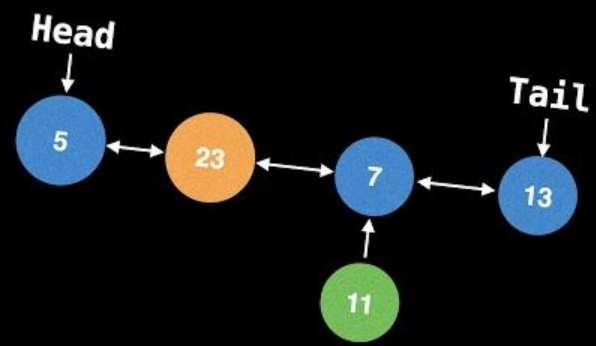
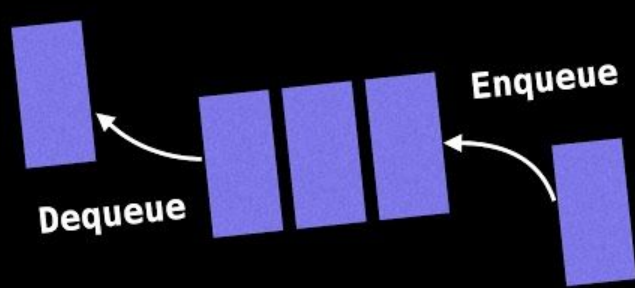
1. Start iterating pixels of image with two “for” loops.
2. If a pixel with interested value is found, start checking its neighboring pixels.
3. If a neighbour is also with the interested value, its neighbour also should be checked as we want to find set of all connected points.



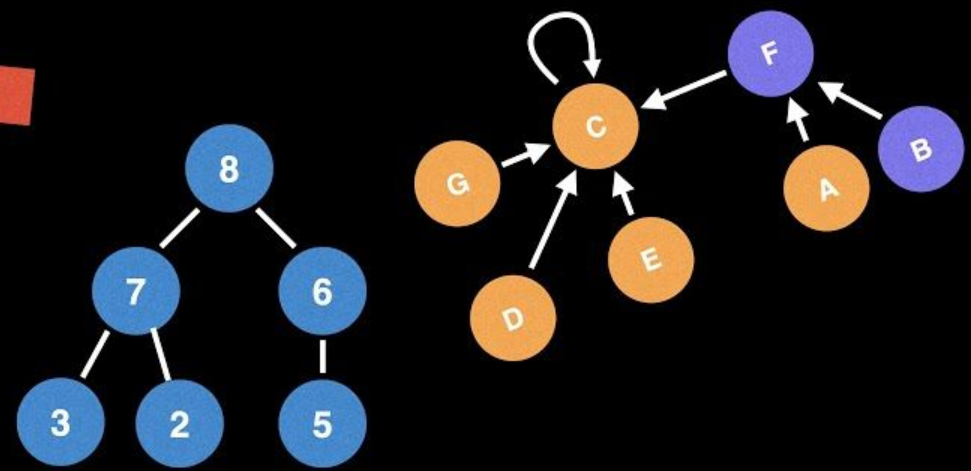
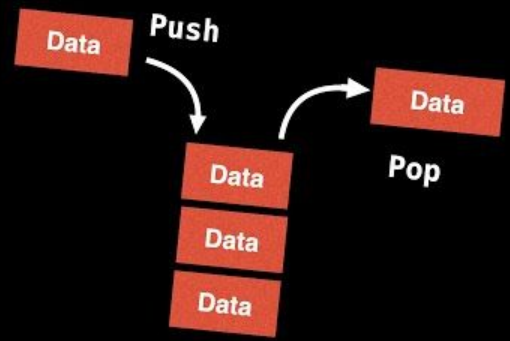
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	255	255
0	0	0	0	0	0	255	255
0	0	0	0	0	0	255	255
0	0	0	0	0	255	255	255
0	0	0	0	0	255	255	255
0	0	0	255	255	255	255	255
0	0	255	255	255	255	255	255
0	0	255	255	255	255	255	255
0	255	255	255	255	255	255	255
0	255	255	255	255	255	255	255
0	255	255	255	255	255	255	255
55	255	255	255	255	255	255	255
55	255	255	255	255	255	255	255
55	255	255	255	255	255	255	255
55	255	255	255	255	255	255	255

## Checking Blobs:

1. Start pixel (0,0)
  2. Interested pixel detected (1,6)
  3. Start checking its neighbours.
- ...
- ..
- .

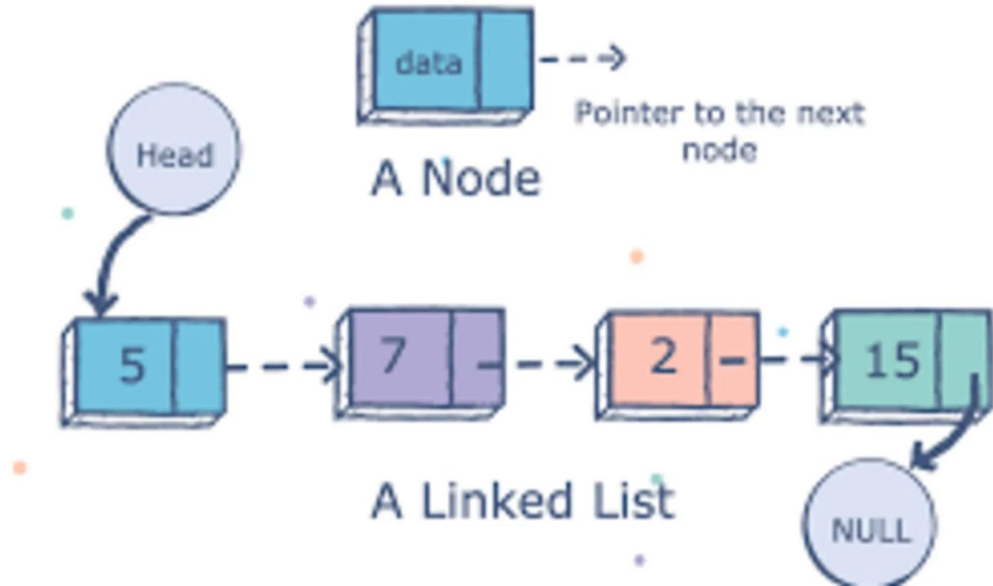


# Data Structures



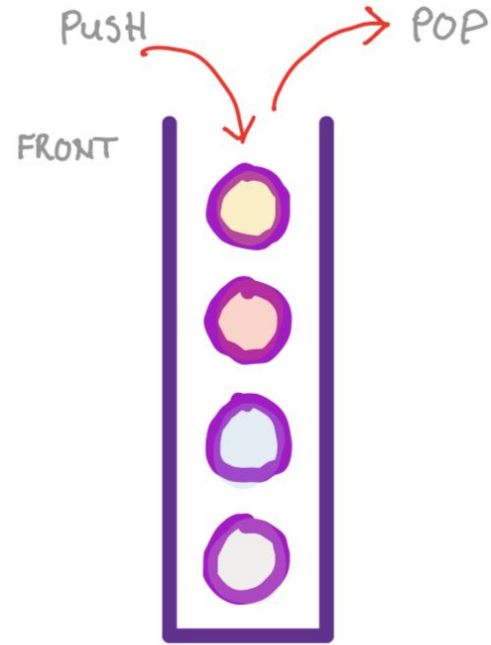
# Linked List

- Arrays - Why Linked Lists?
- Pointers
- Node



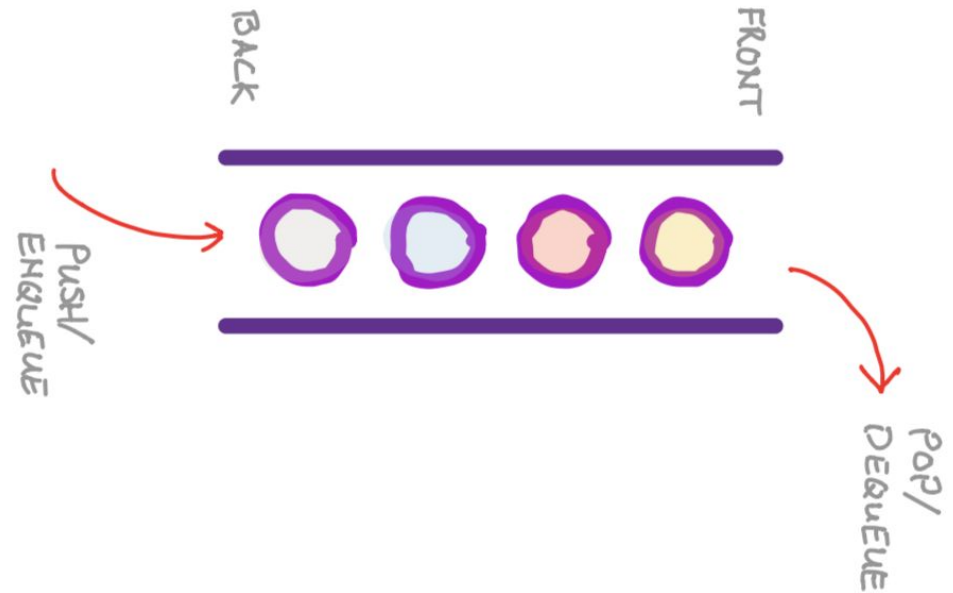
# Stack

Last-in-first-out (LIFO)



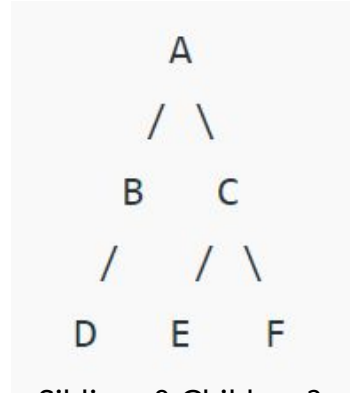
# Queue

First-in-first-out (FIFO)



[Visualizer](#)

# BFS & DFS



Siblings & Children ?

[Visualizer](#)

## BFS

- Breadth First Search
- **Siblings are visited first**, then the children.
- Uses **Queue** data structure.
- Guarantees **shortest path**

A, B, C, D, E, F

## DFS

- Depth First Search
- **Childrens are visited before the siblings.**
  - **Stack** is used.
- Doesn't guarantee the shortest path

A, B, D, C, E, F / A, C, F, E, B, D

Questions?

---



# References

[OpenCV Blob Detecion](#)

[Blob Detecion Area, Circularity](#)

[Stacks and Queues](#)

[Implementing Stacks & Queues using list in Python](#)

[BFS Implementation for Trees in Python](#)

[DFS Implementation for Trees in Python](#)

[Pathfinding algo Visualizer](#)