gextract: Graph Construction from Images

Clara Nguyễn

ECE 572 – 2019/12/03

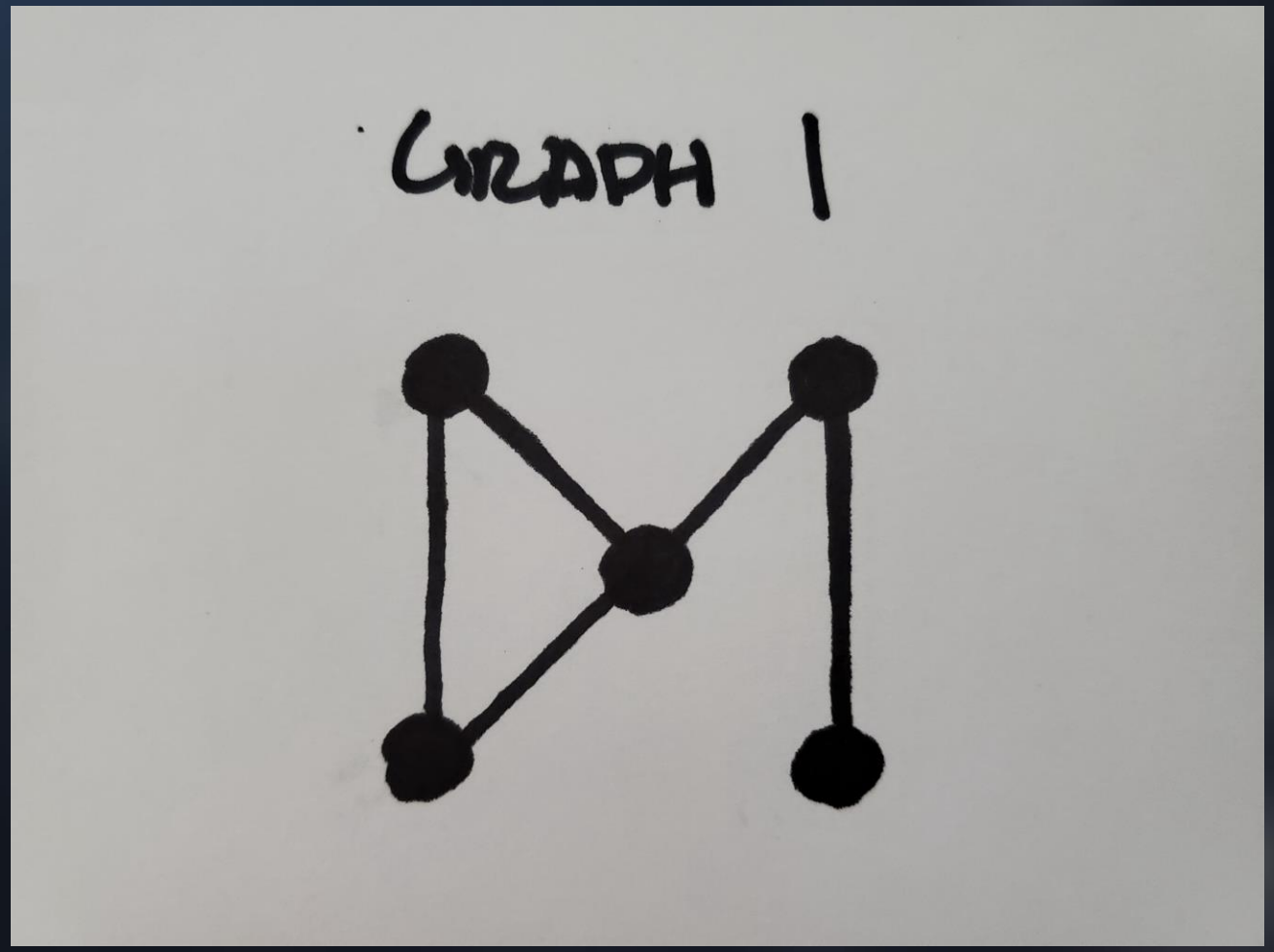
First off…

- This is a speedy talk. If you want the full details:

  https://tiny.utk.edu/talk4

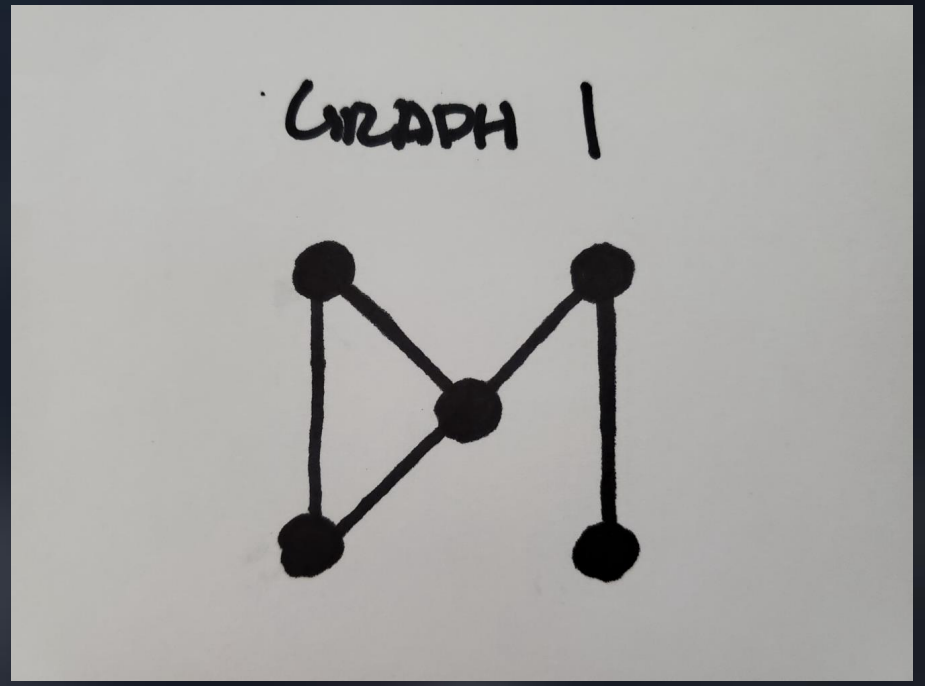
The Problem

This is a graph…



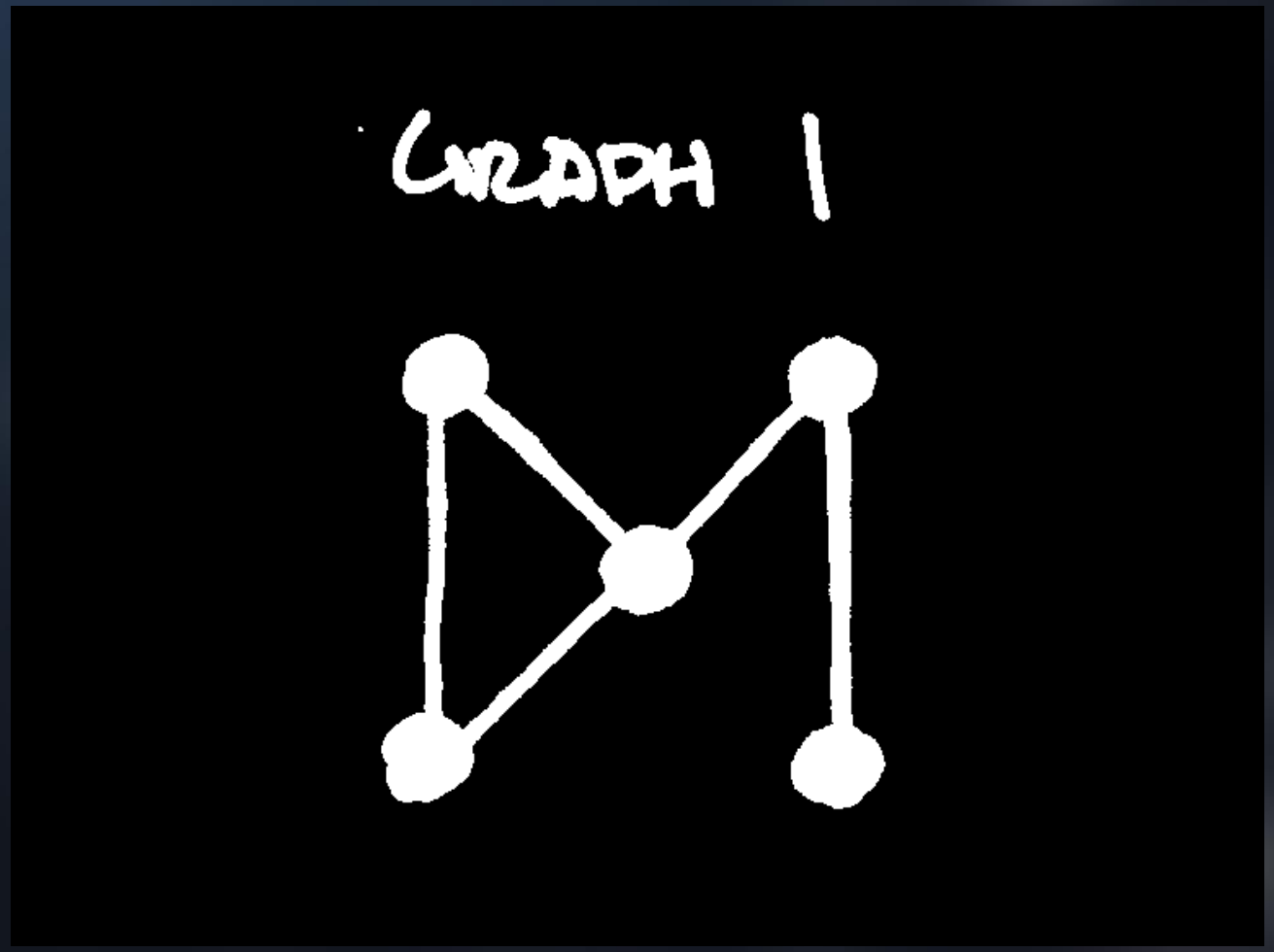
This graph is…

- A picture taken on a smartphone.
- Lacking in any information other than nodes and edges.
- Not directly readable as a graph to your average program…

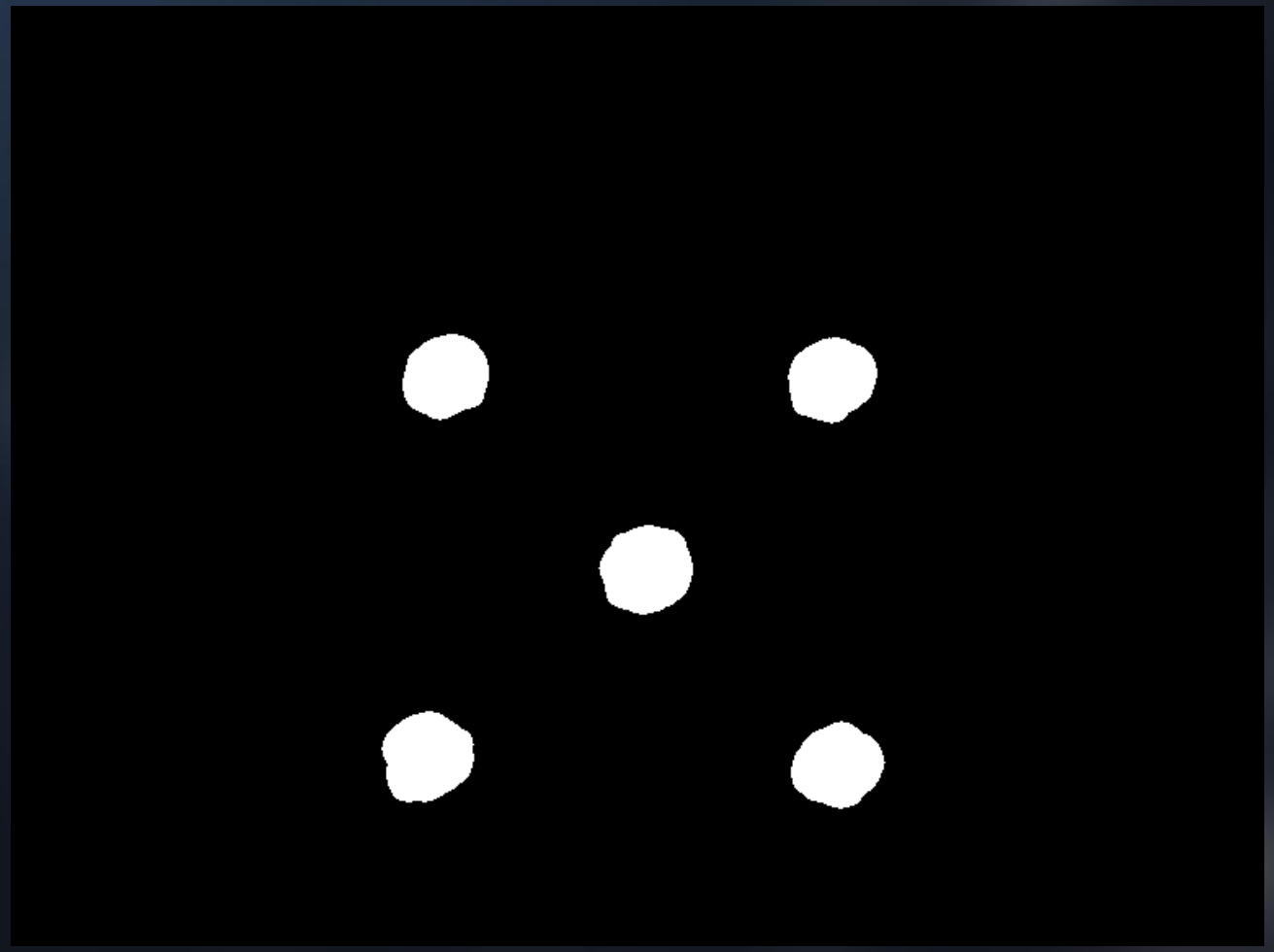


Let’s change that…

Binarisation and Inversion



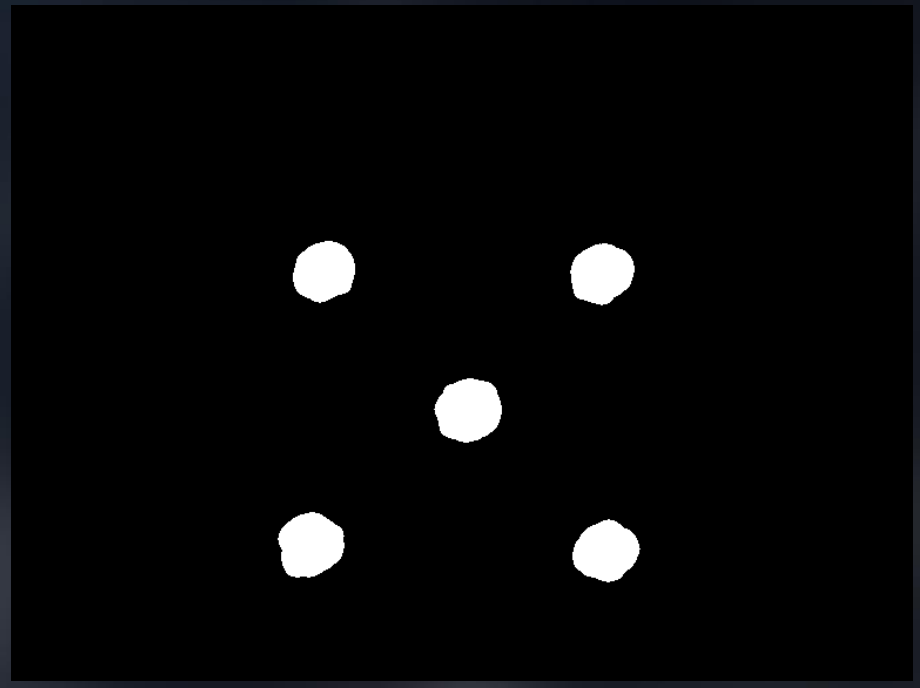
Applied Morphology: Opening



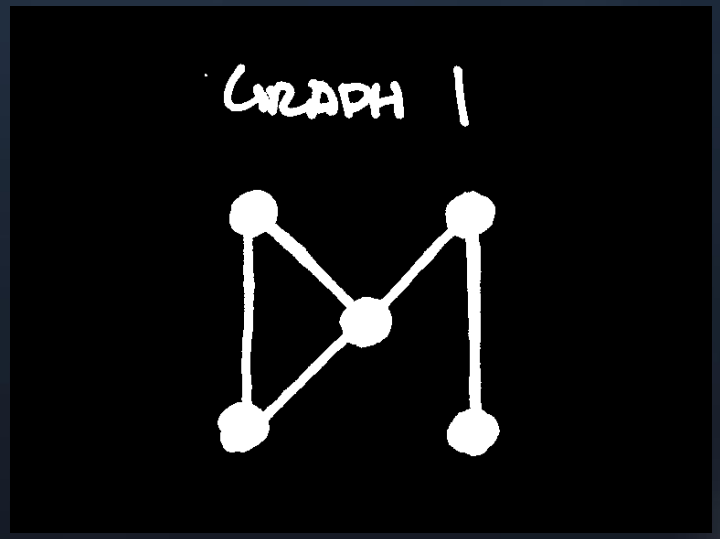
Nodes extracted

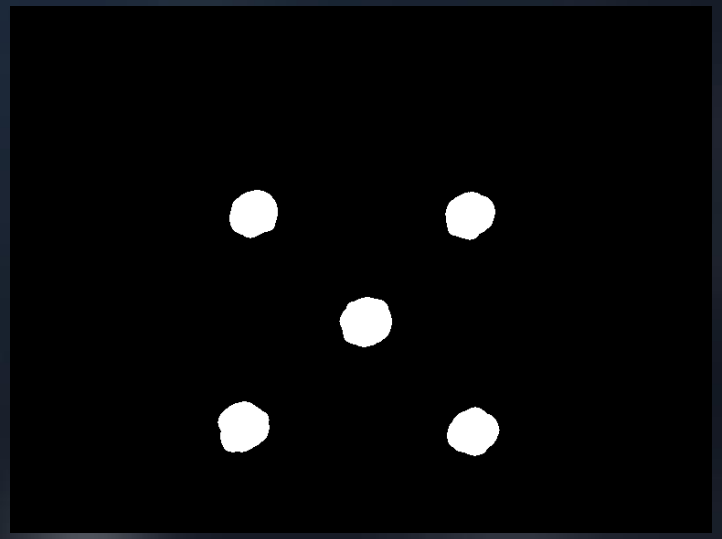
- Can extract node “objects” via **Connected Components**.

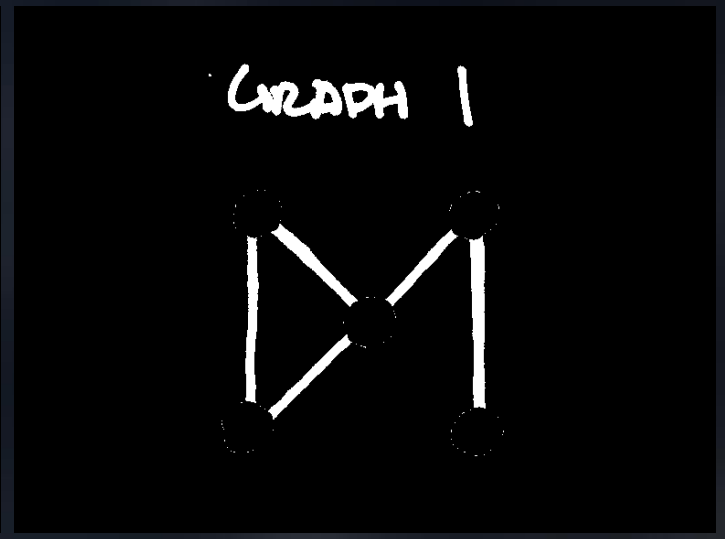
- What next? We can extract edges by taking difference of this and binarized + inverted image.



Difference Computation

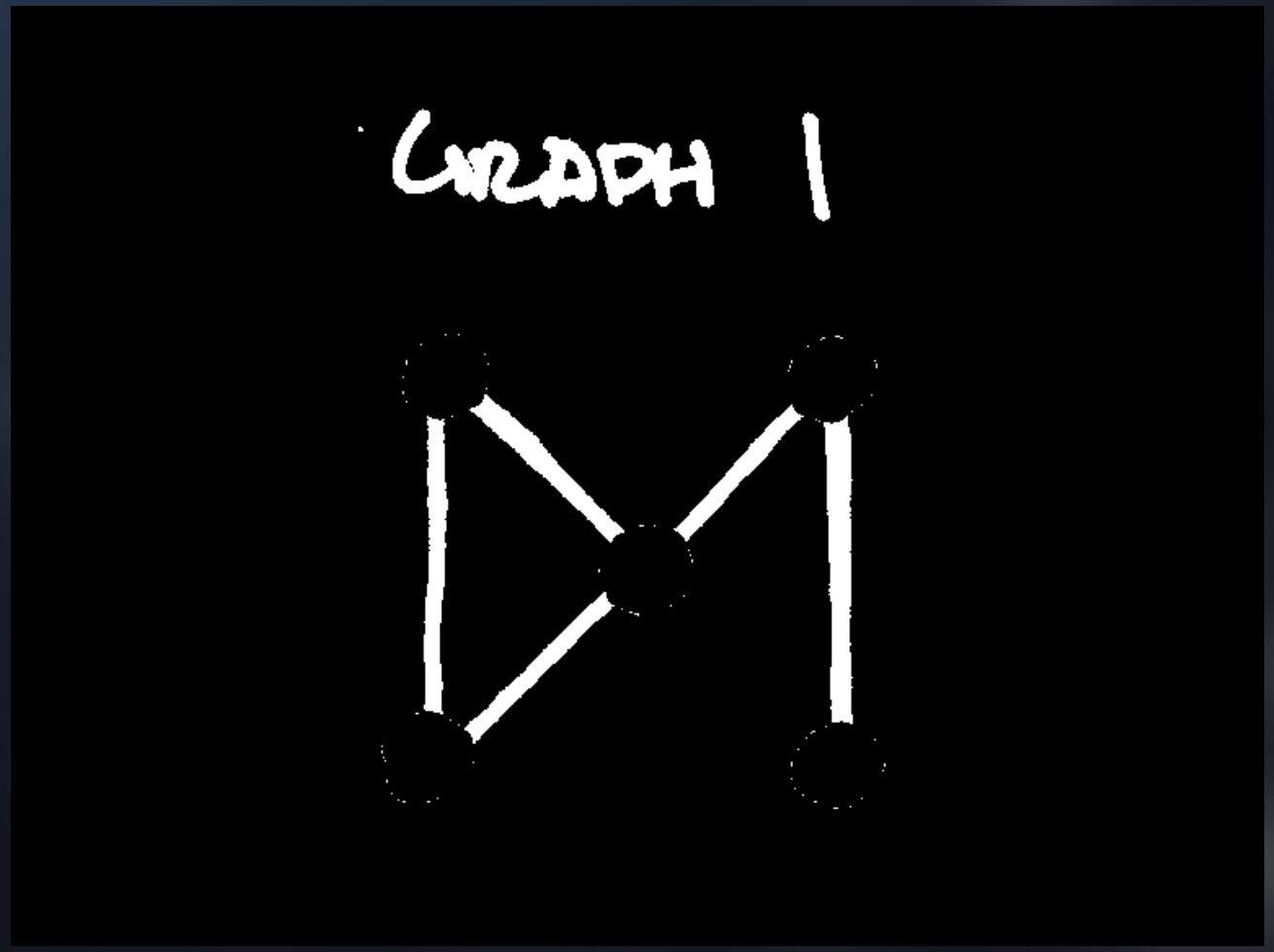






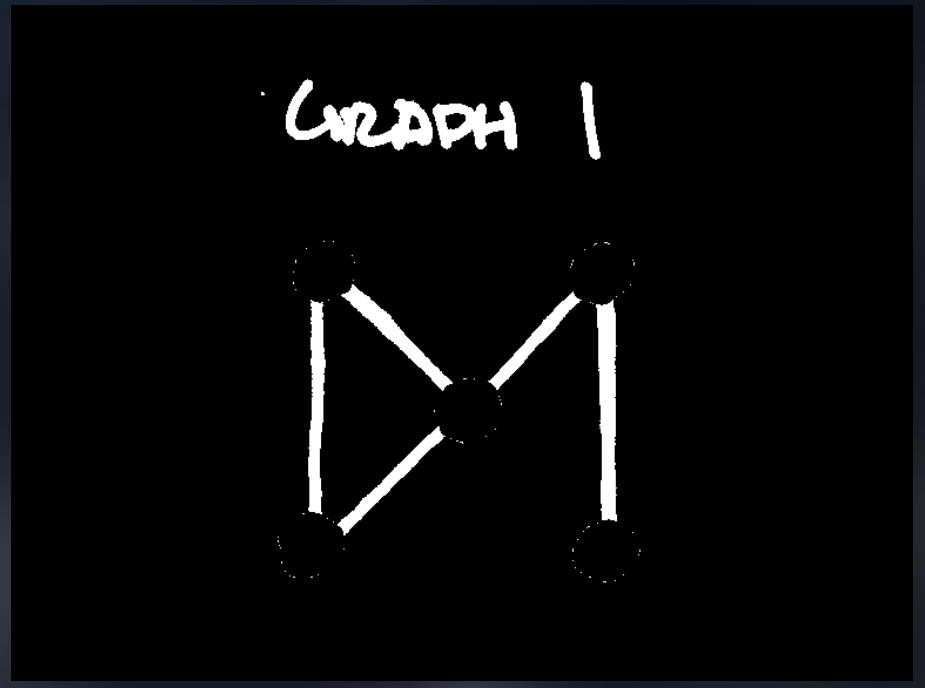
Original XOR Nodes = Edges?

Difference Result. What's wrong?



Difference Result. What’s wrong?

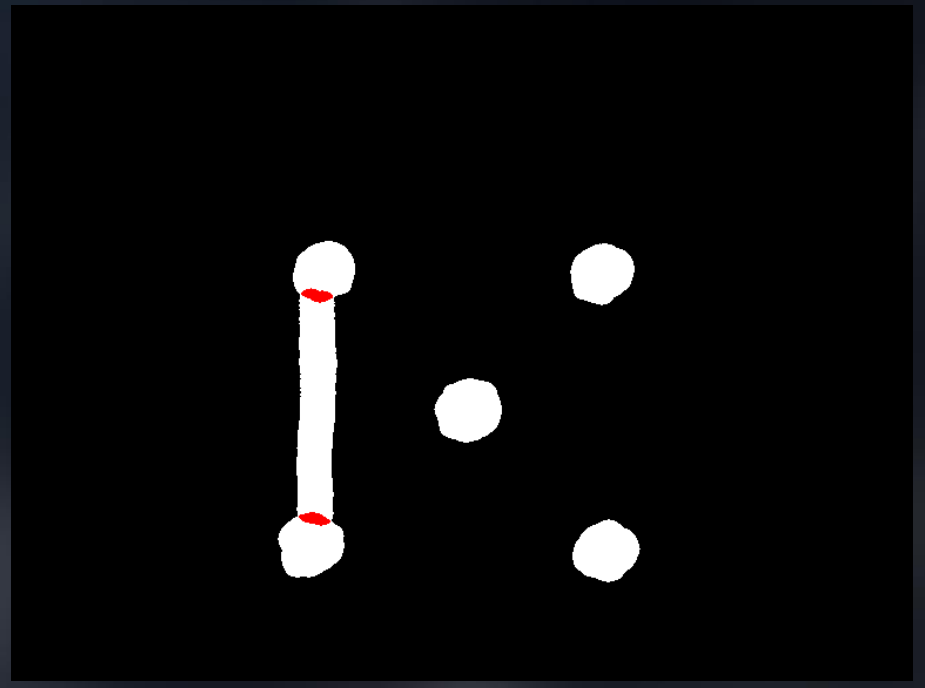
- Text at the top is still there
- Minor fragments where nodes used to be still appear.
- These issues don’t matter. Again, use **Connected Components**. Keep objects with more than 100 pixels.



Connect the graph

- Extract each edge identified by Connected Components and put into their own subimage. Then **dilate** each.

- Go through each edge sub-image and check which nodes they meet. Use this to “connect” the graph.



Applications of this

- We can take this data and perform graph algorithms on them.
- We can even run “shortest path” algorithms like **Dijkstra’s Algorithm** if we use the pixels of each edge as a weight.

Web App Demo

gextract: Graph Construction from Images

Clara Nguyễn

ECE 572 – 2019/12/03