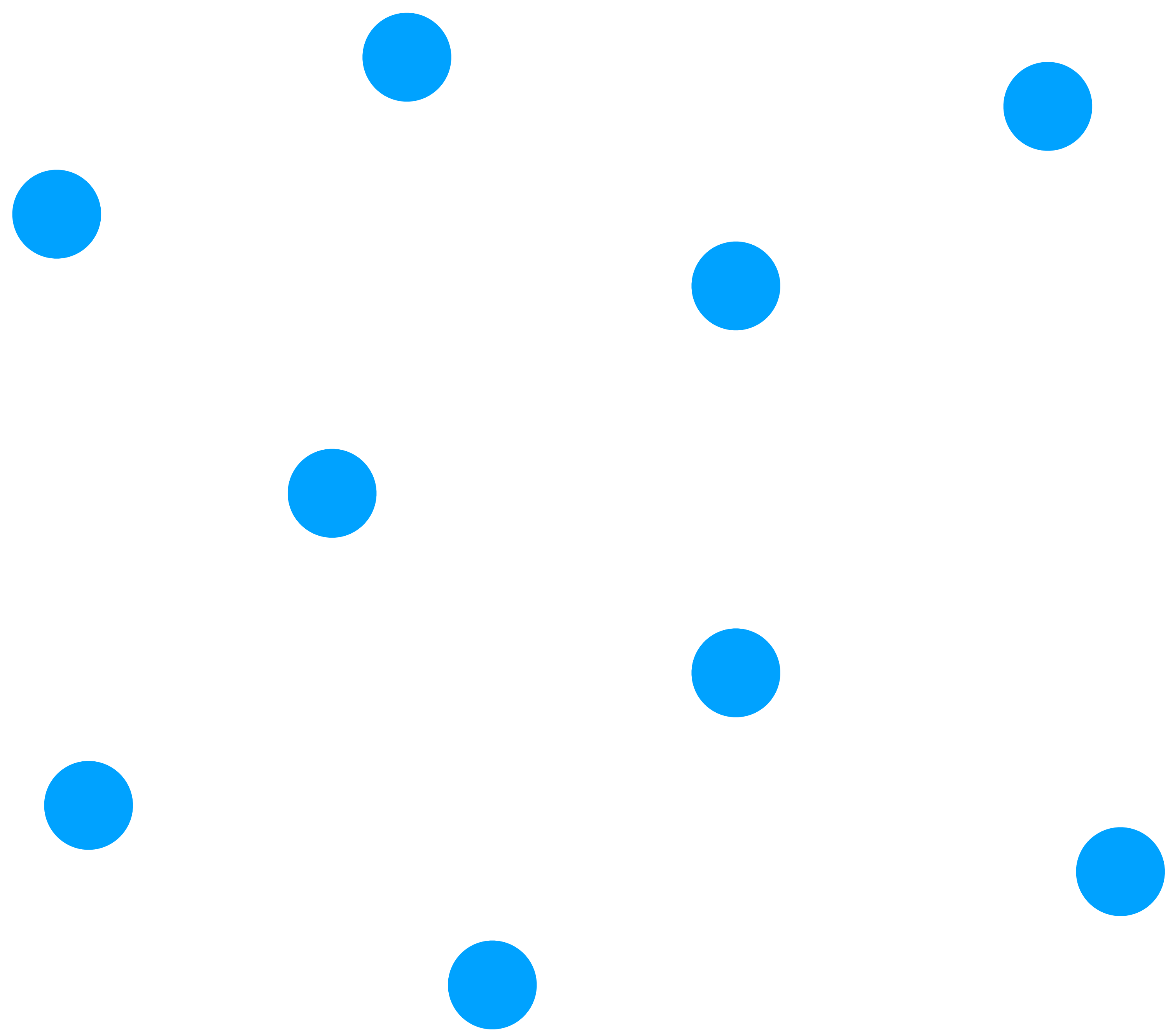
Nearest Neighbor using KDTrees

Nearest Neighbor

- K-nearest neighbor is a simple classifier
- But it’s slow!

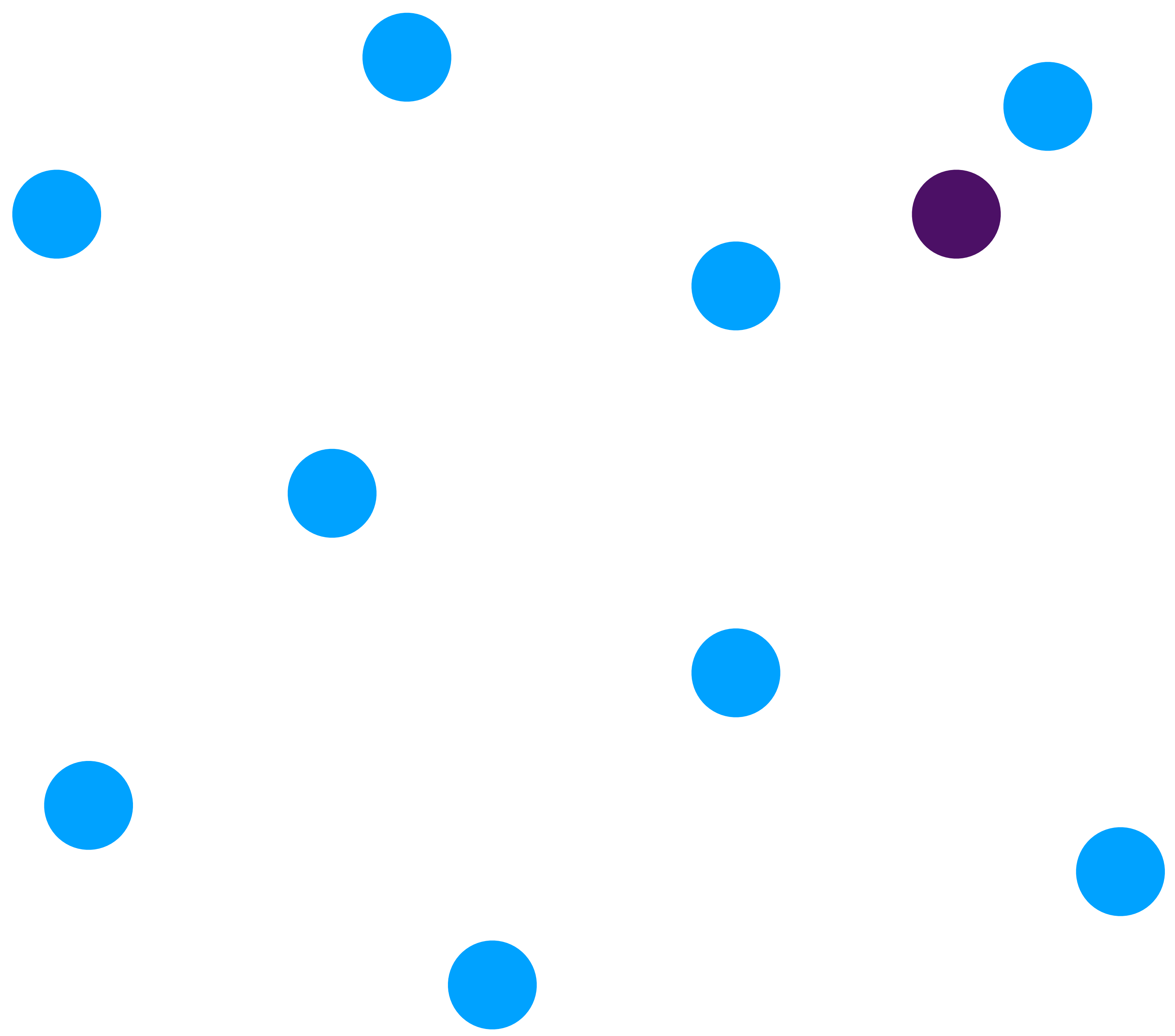


Nearest Neighbor

- K-nearest neighbor is a simple classifier

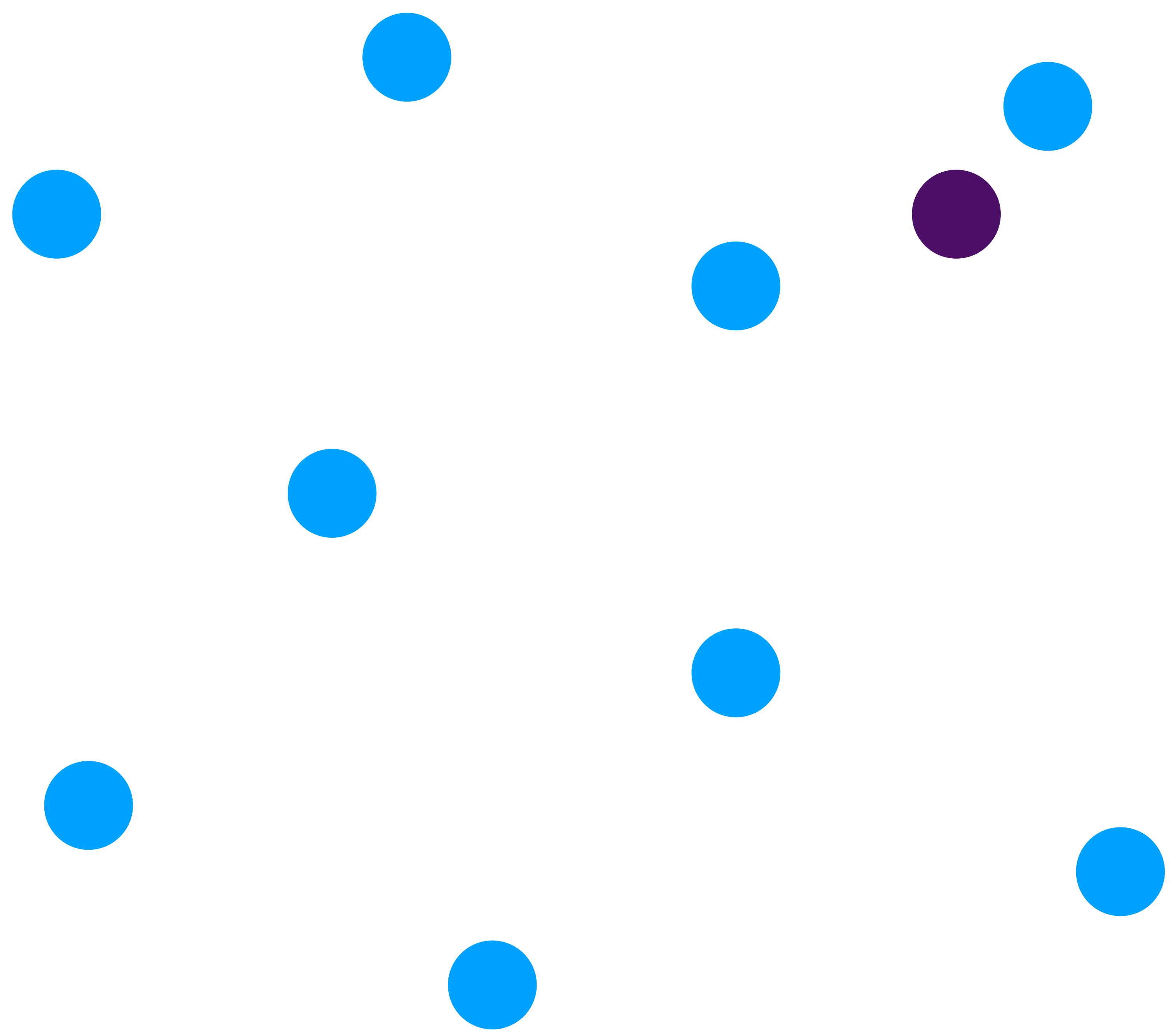
- But it’s slow!

- How many comparisons would we need to make to find the nearest neighbor for the **purple** point?



Accelerating NN

- What if we could use the results of earlier comparisons to avoid doing later comparisons?
- If we already found a pretty close neighbor, an we skip looking at far away points?
- How can we do this without already known which points are “far away”?
- Idea: precompute some spatial information

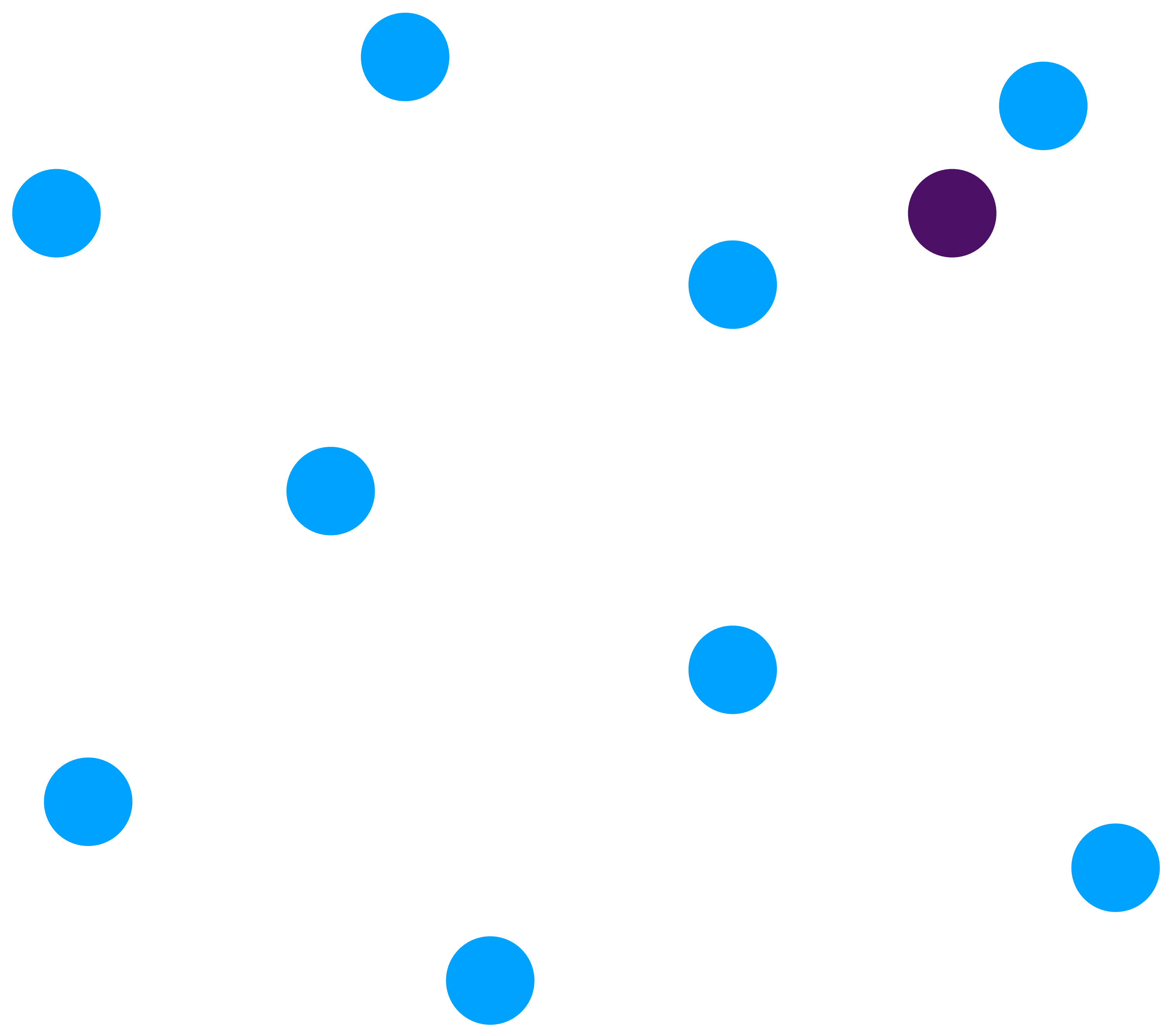


KDTrees

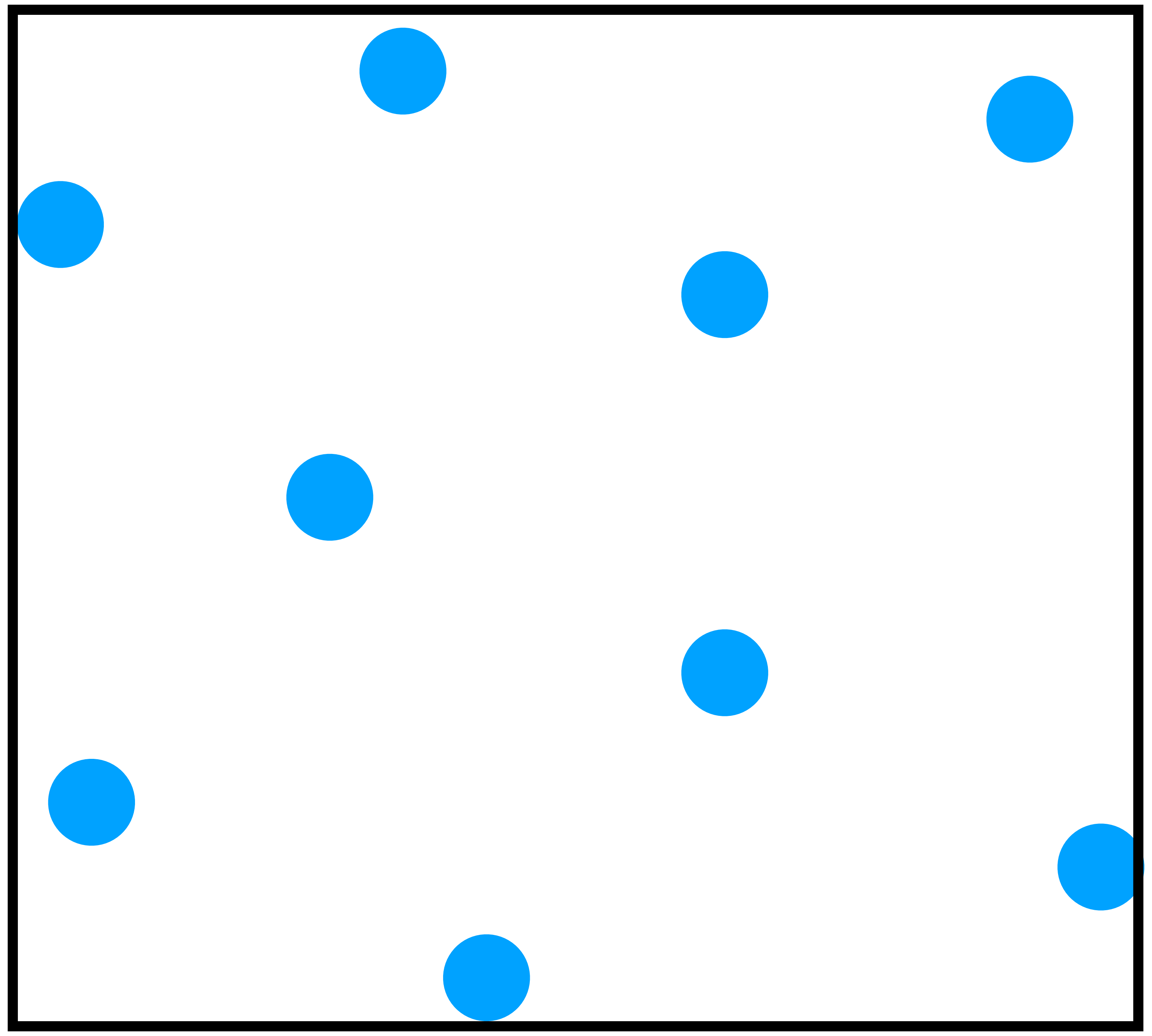
- A *spatial acceleration structure*

- One of many such trees: octtrees, ball trees, BSP trees

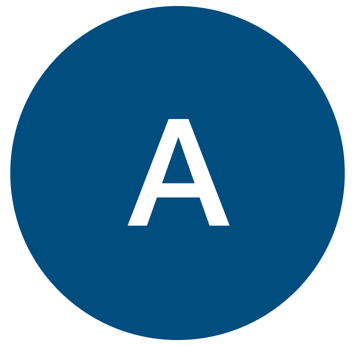
- Hierarchical breakdown of space.

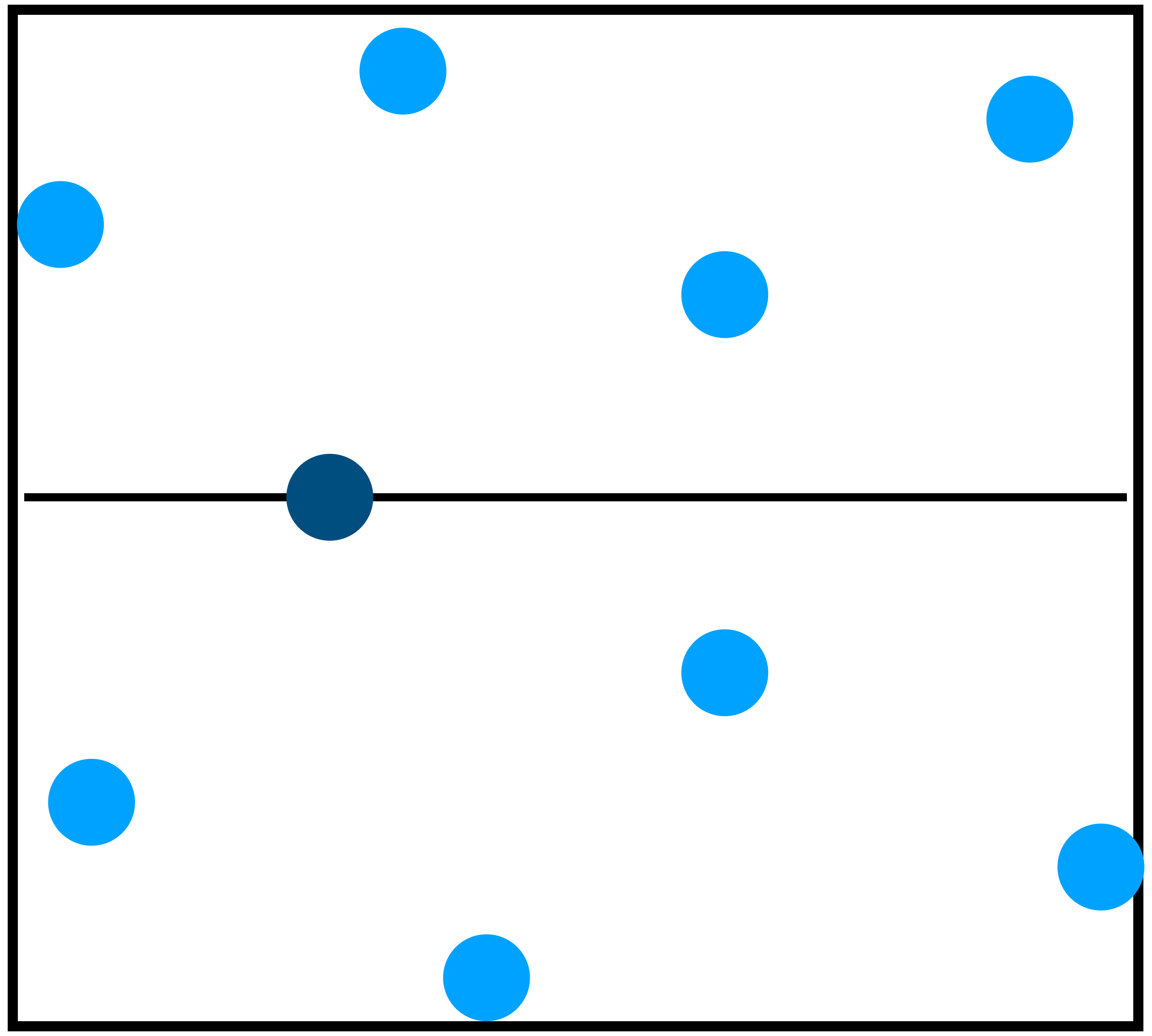


KDTrees

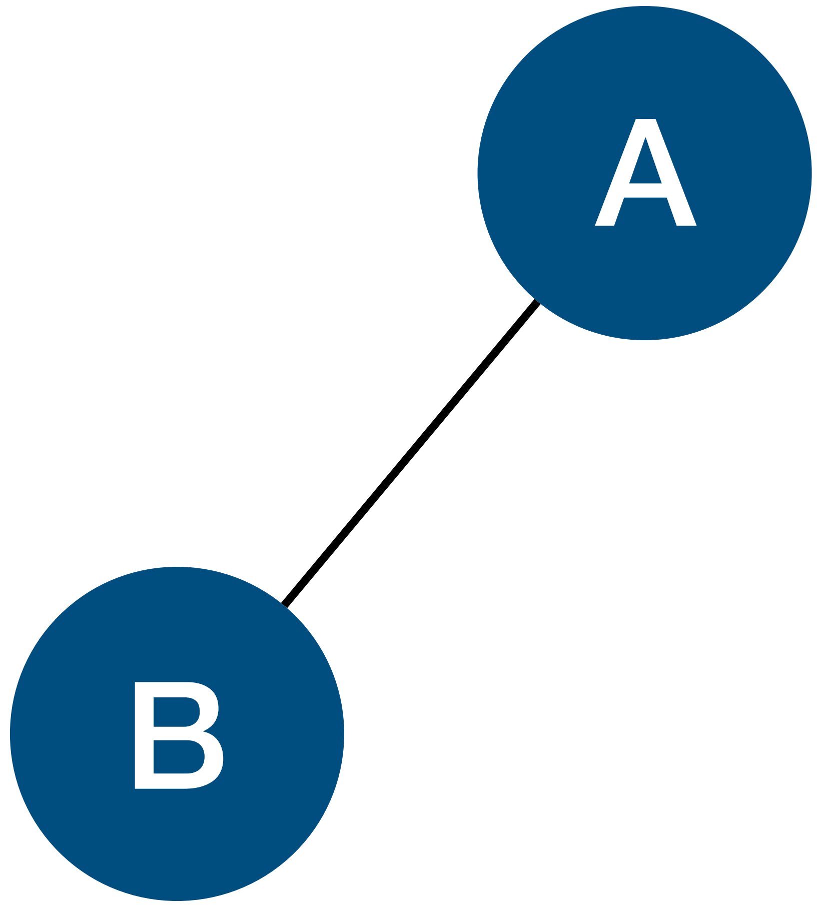


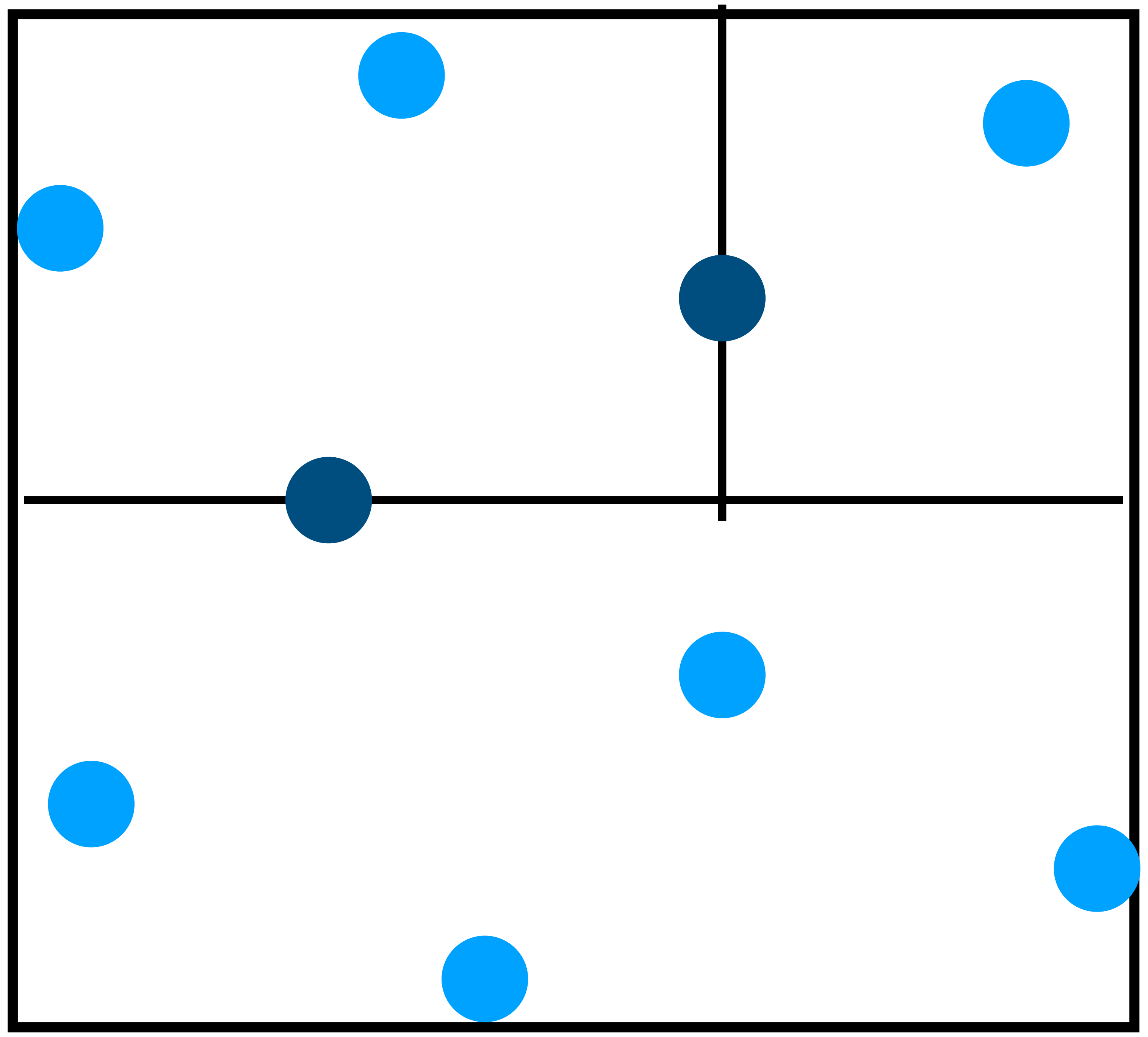
KDTrees



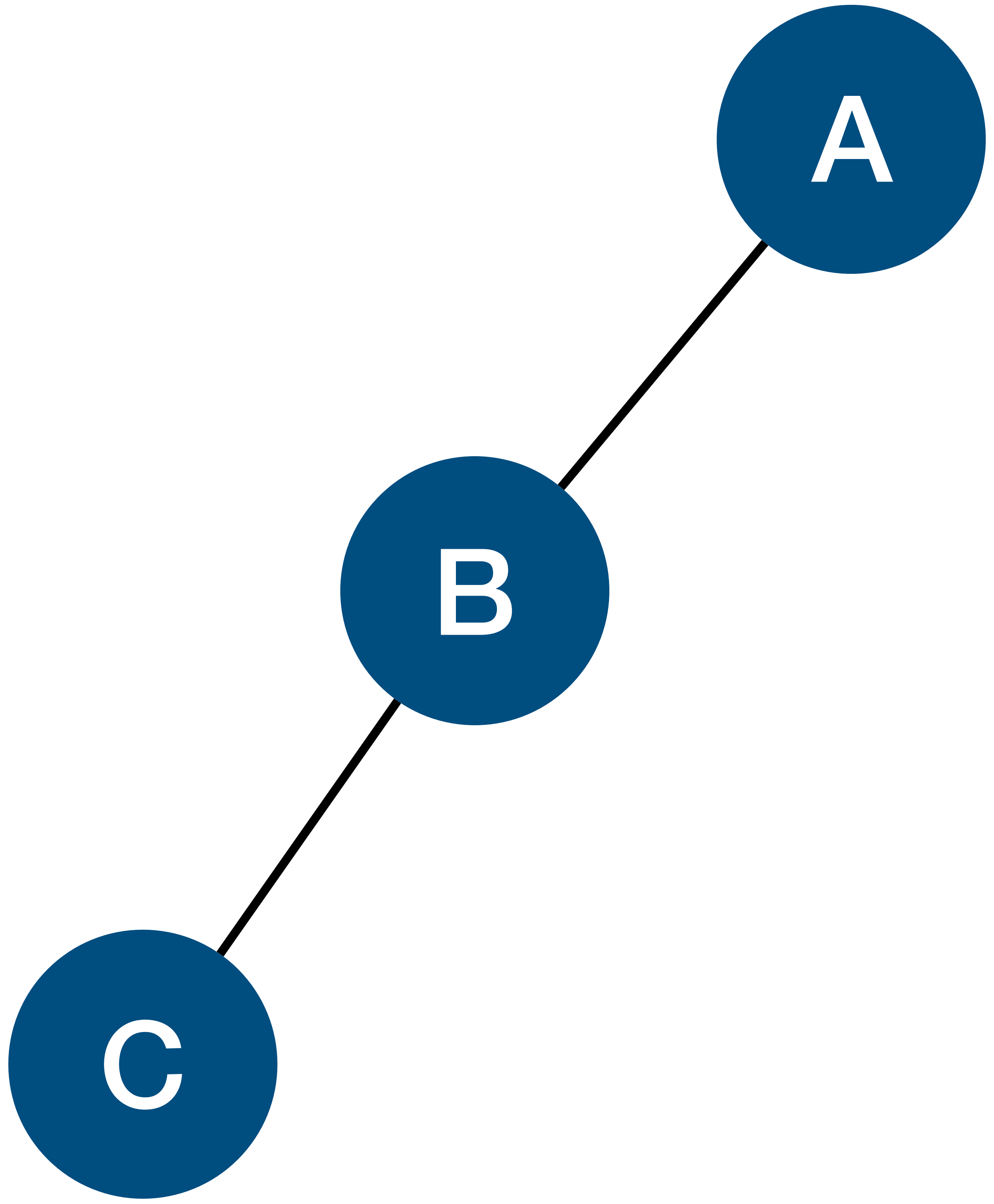


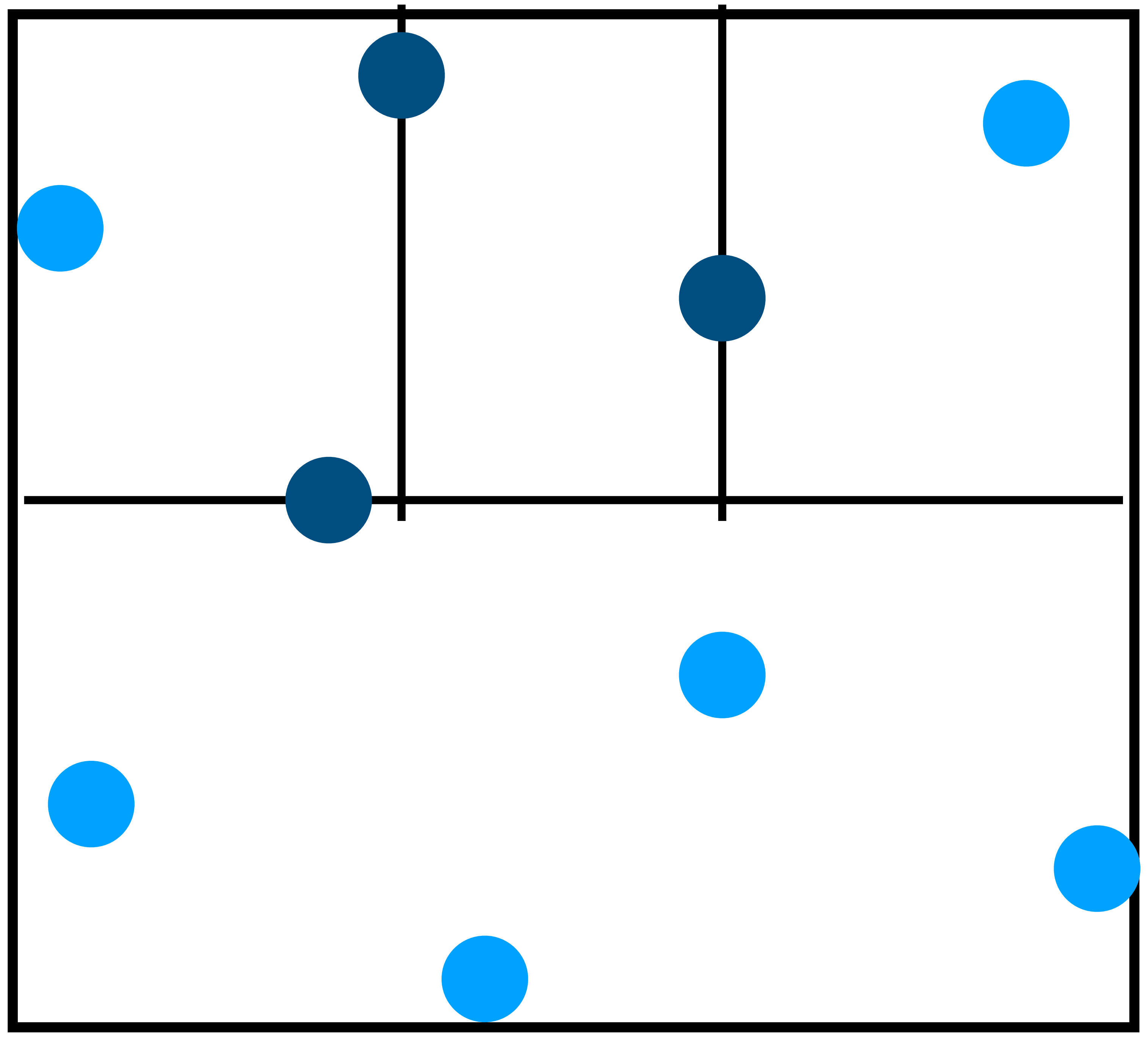
KDTrees



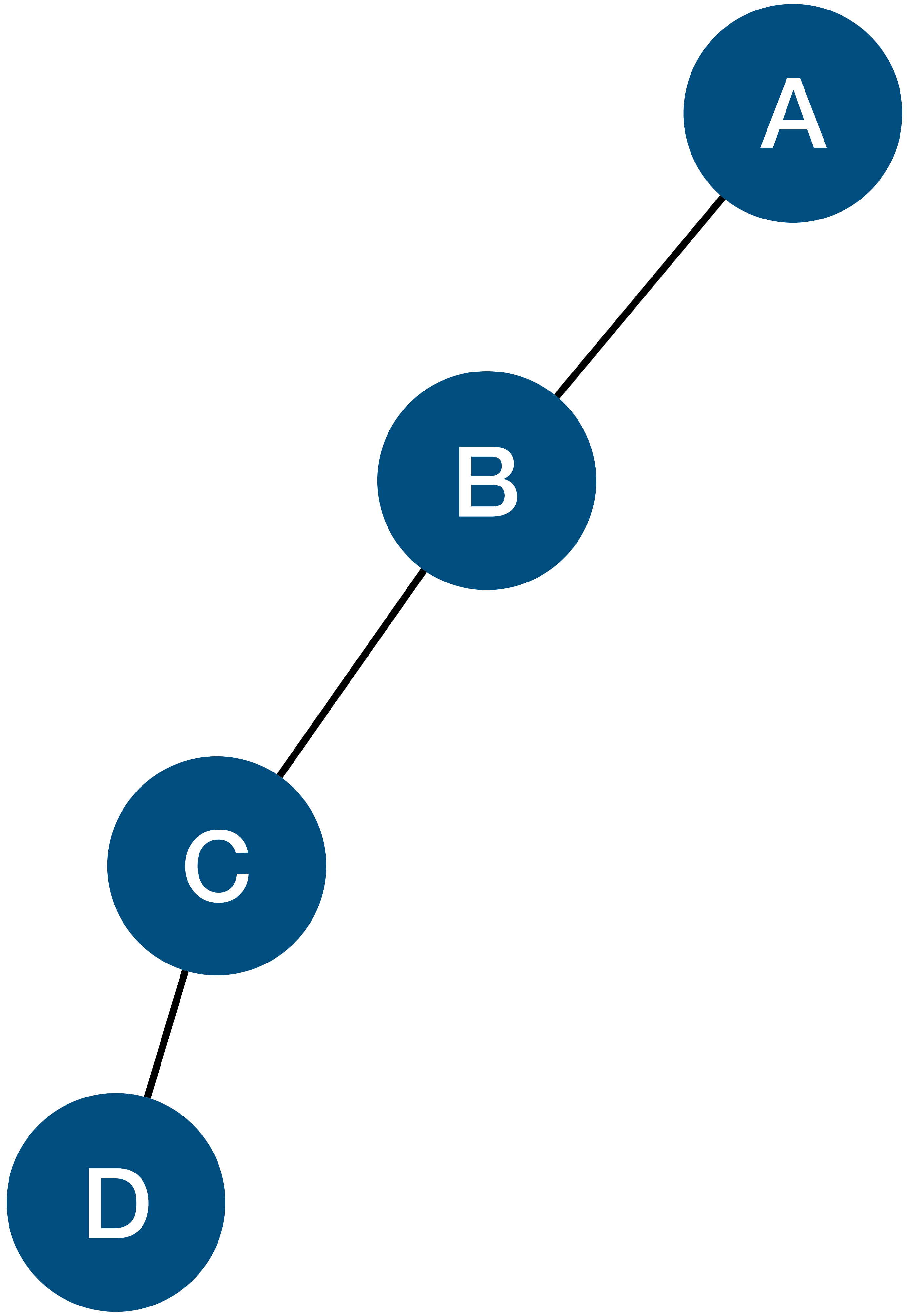


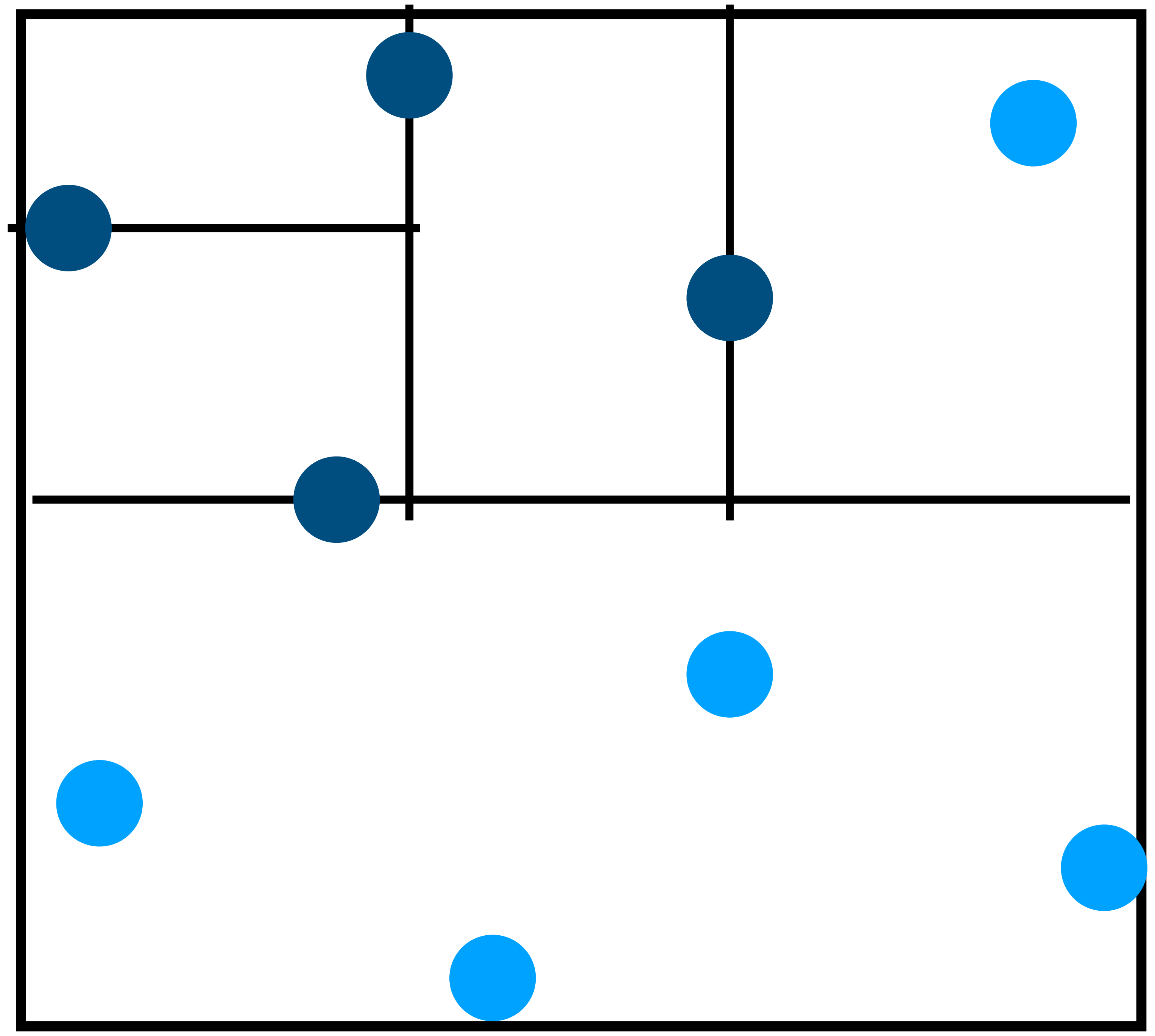
KDTrees



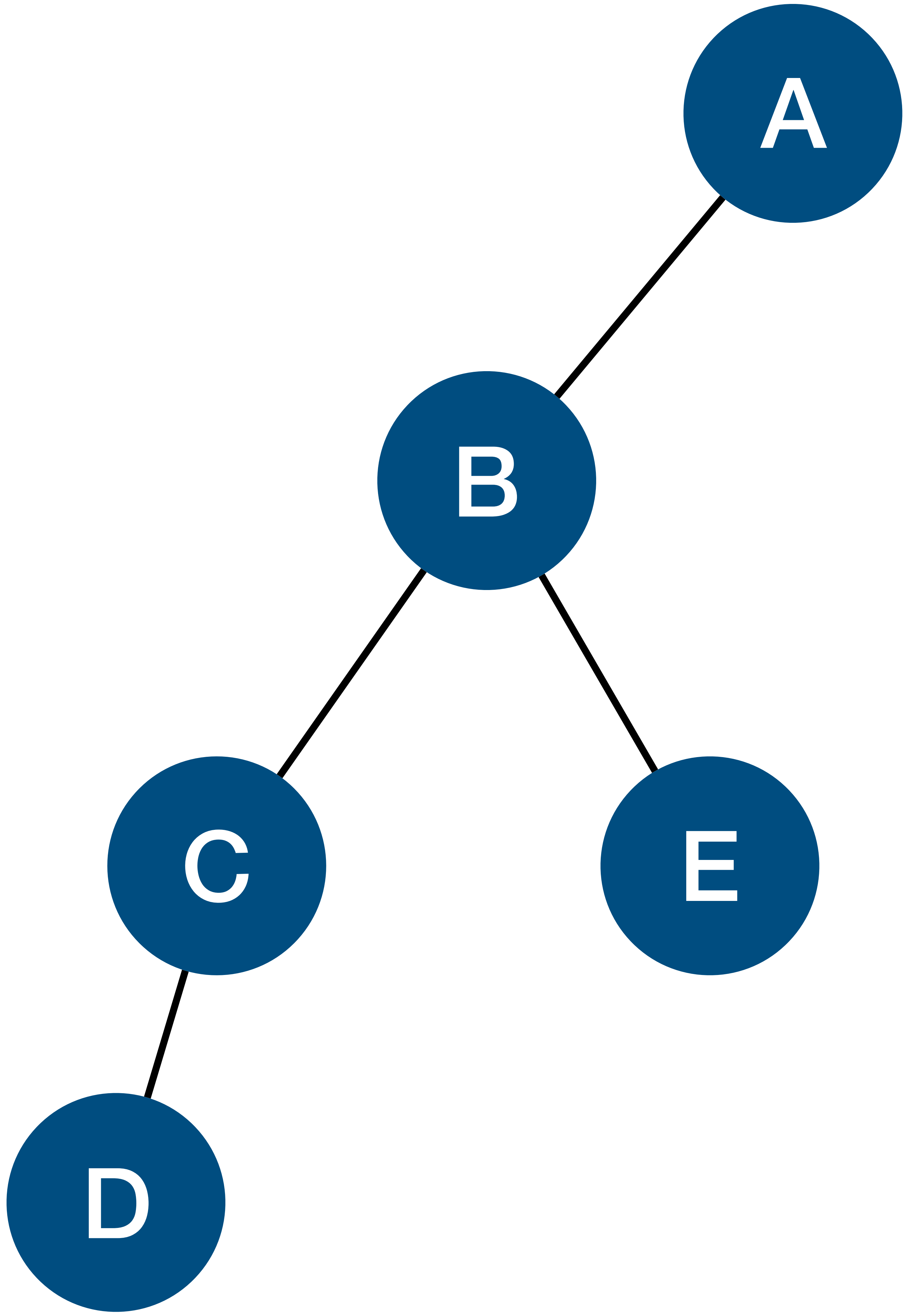


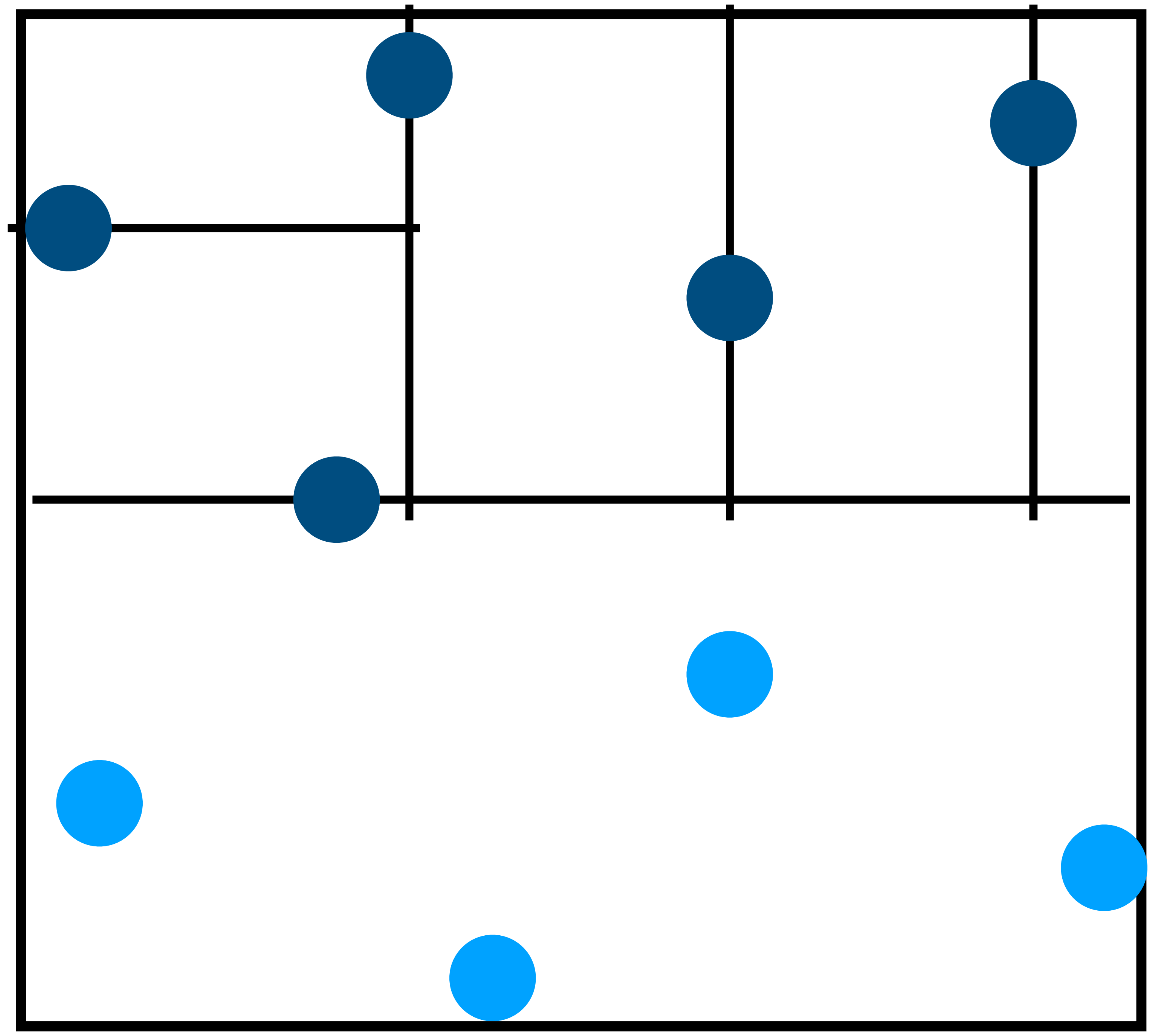
KDTrees



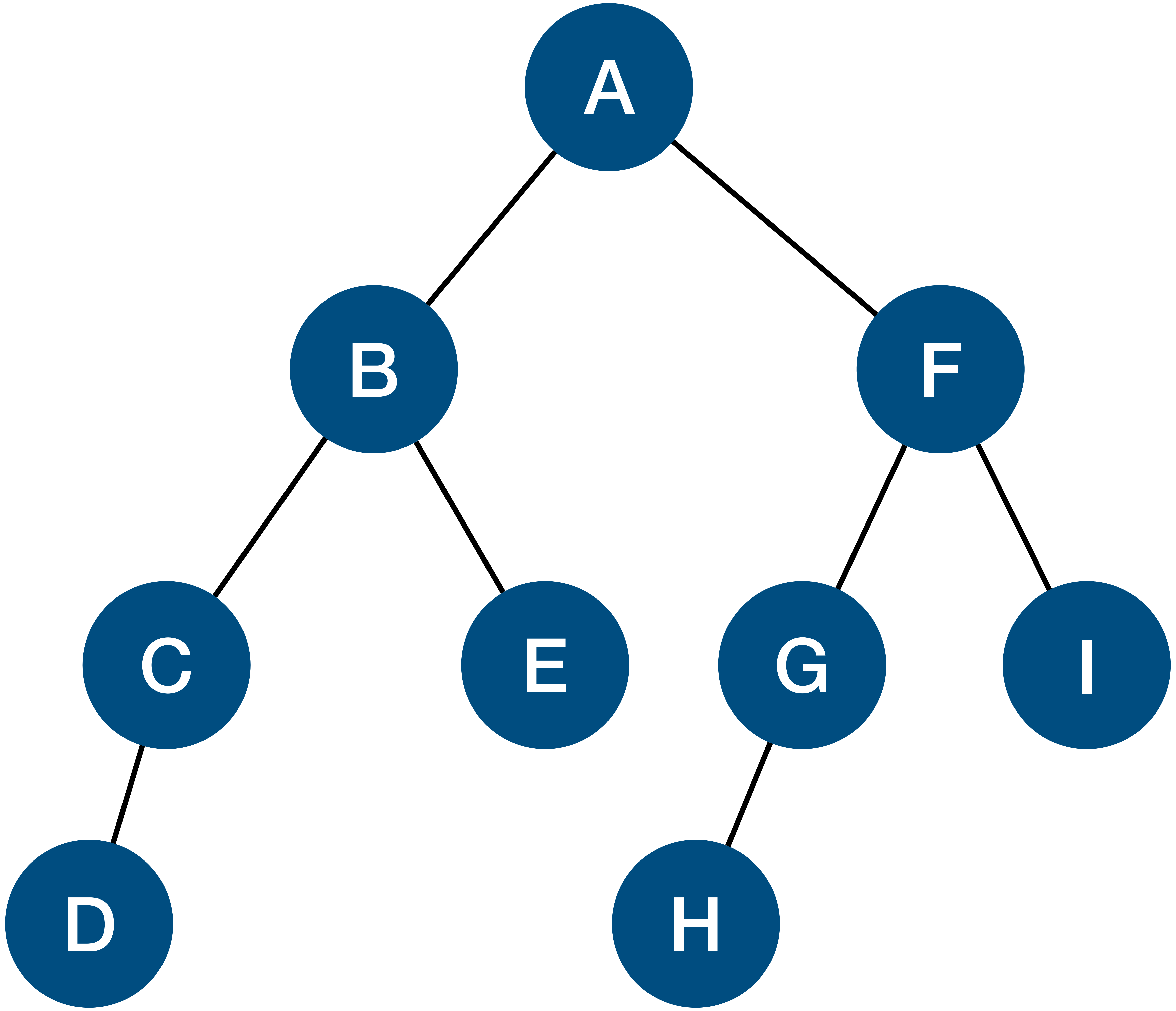


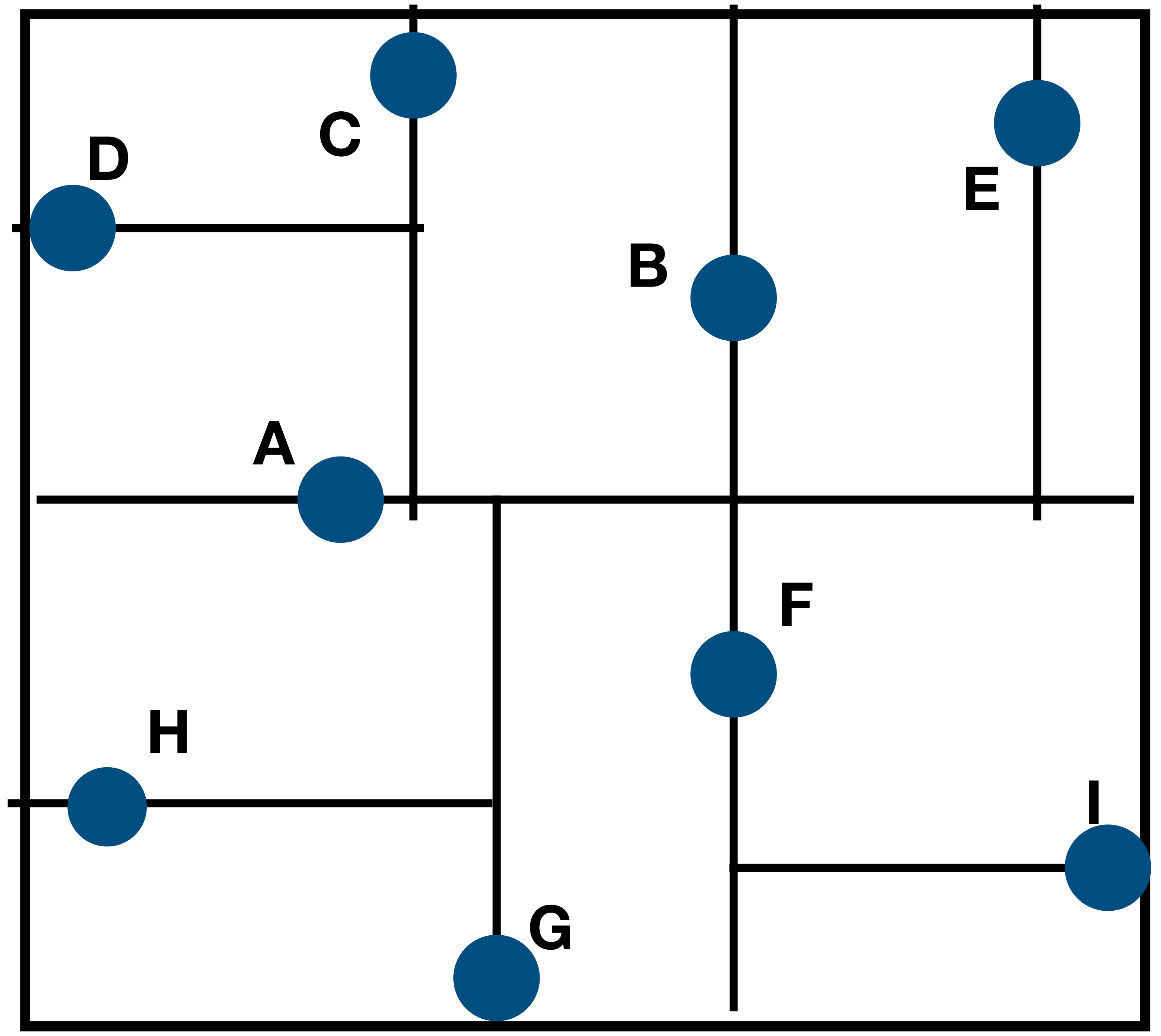
KDTrees



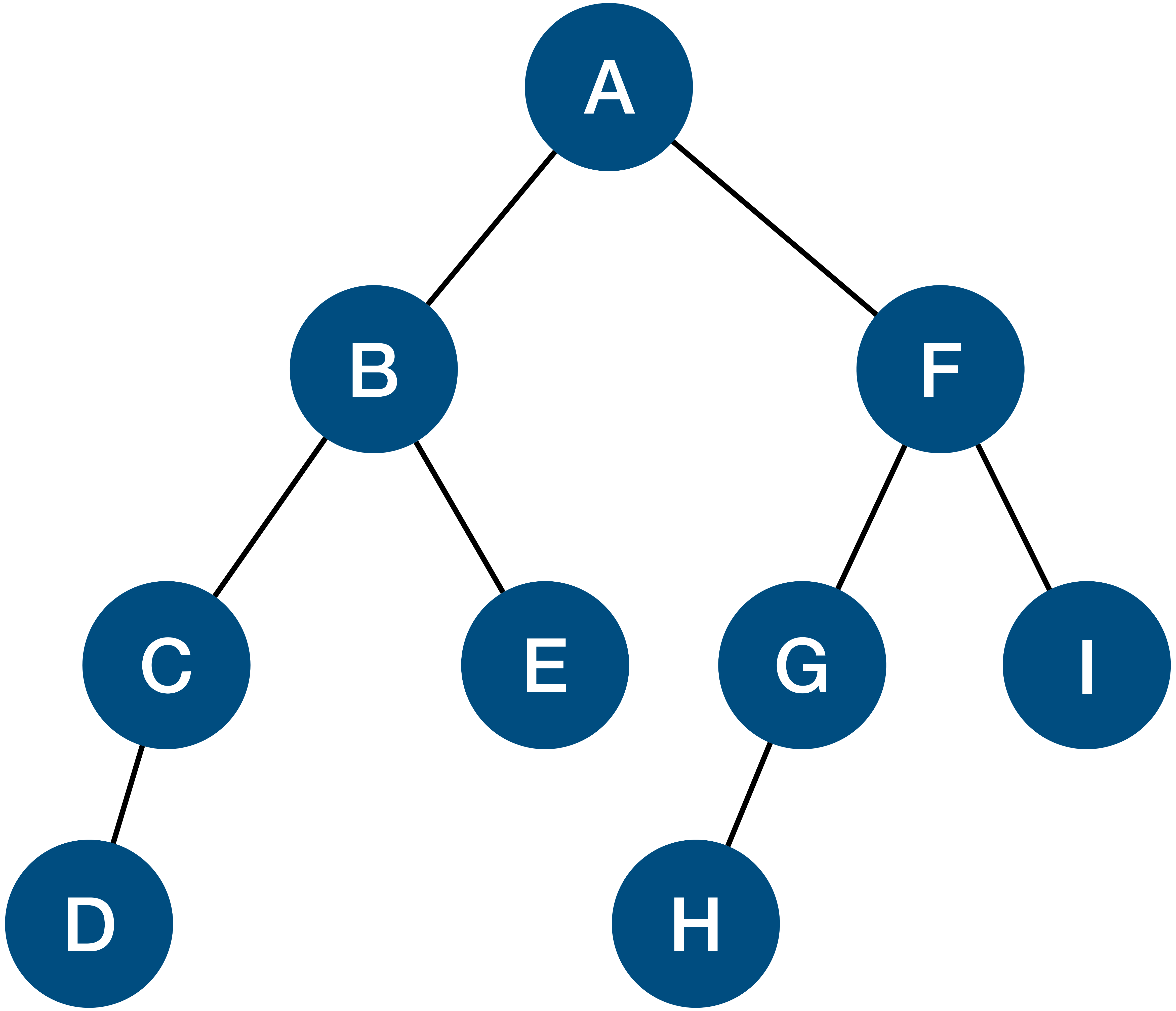


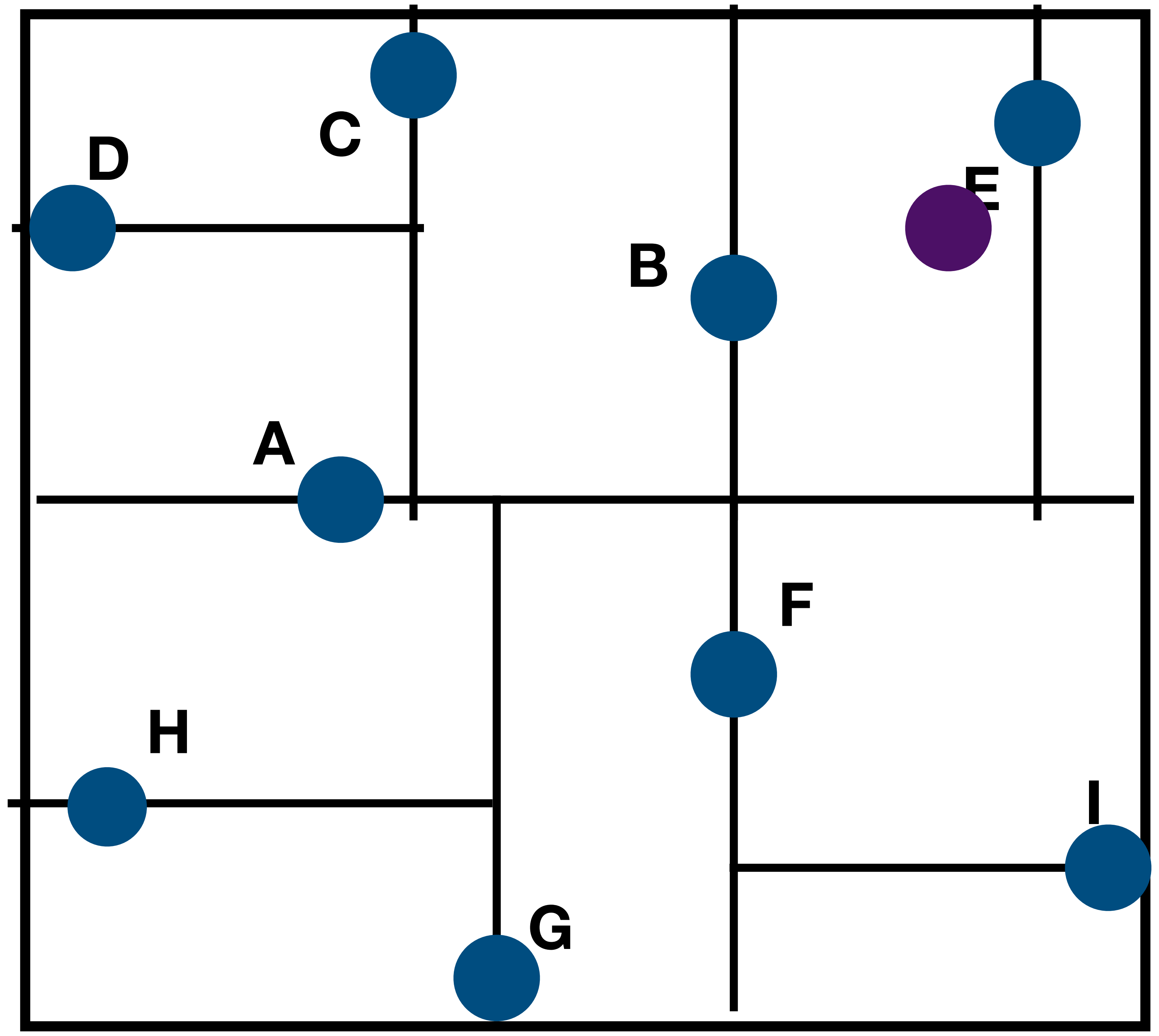
KDTrees



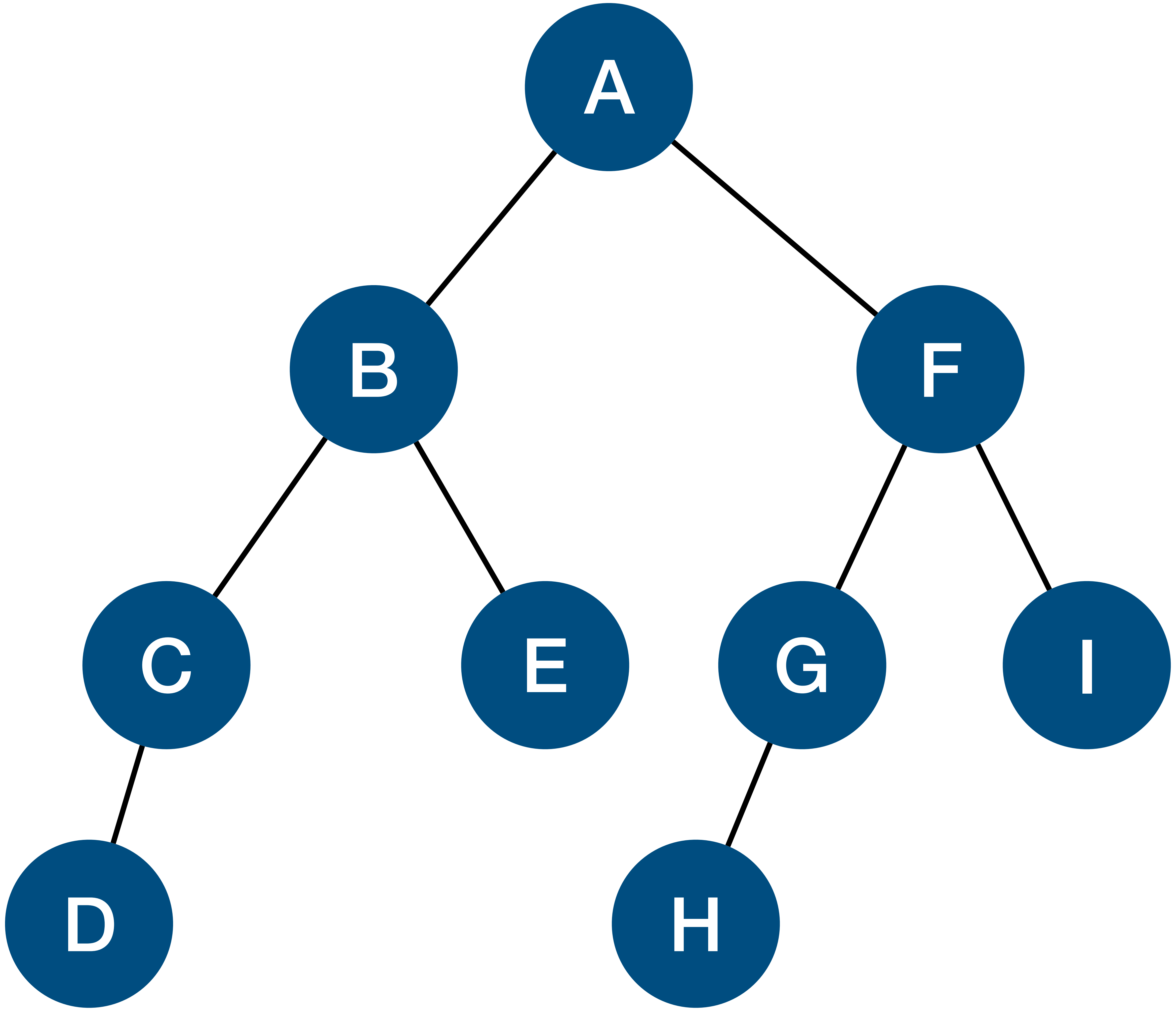


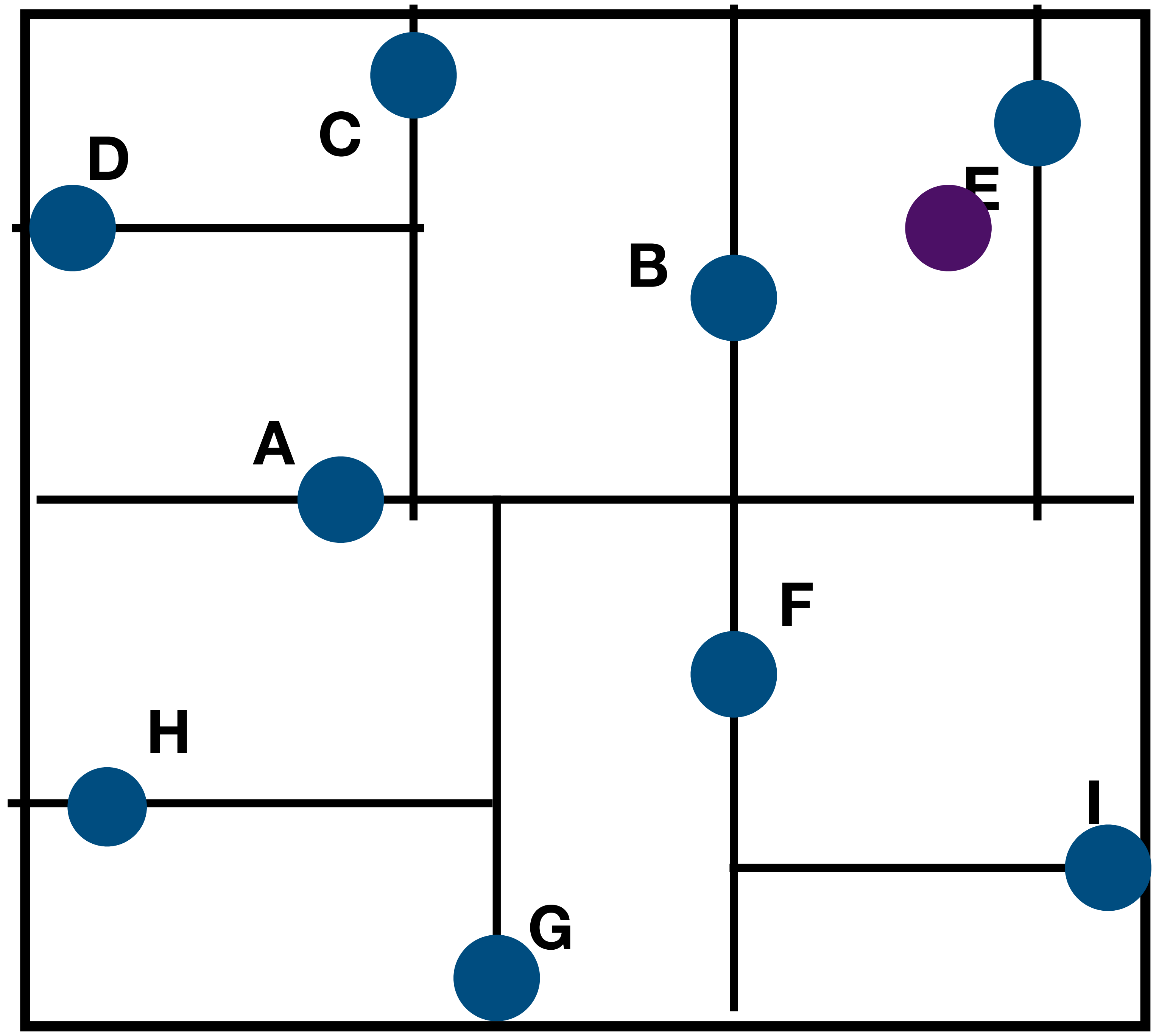
Using a KDTree



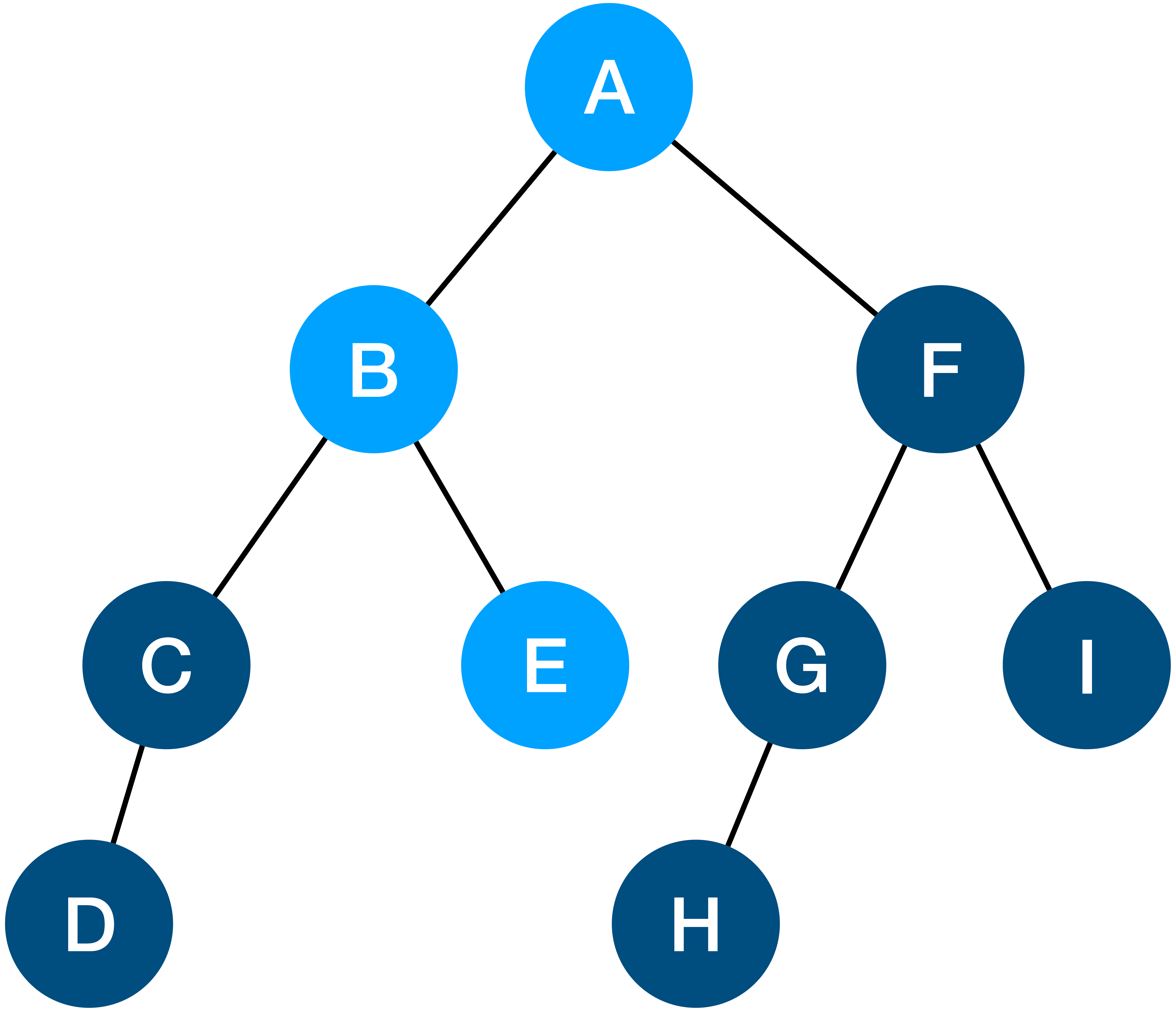


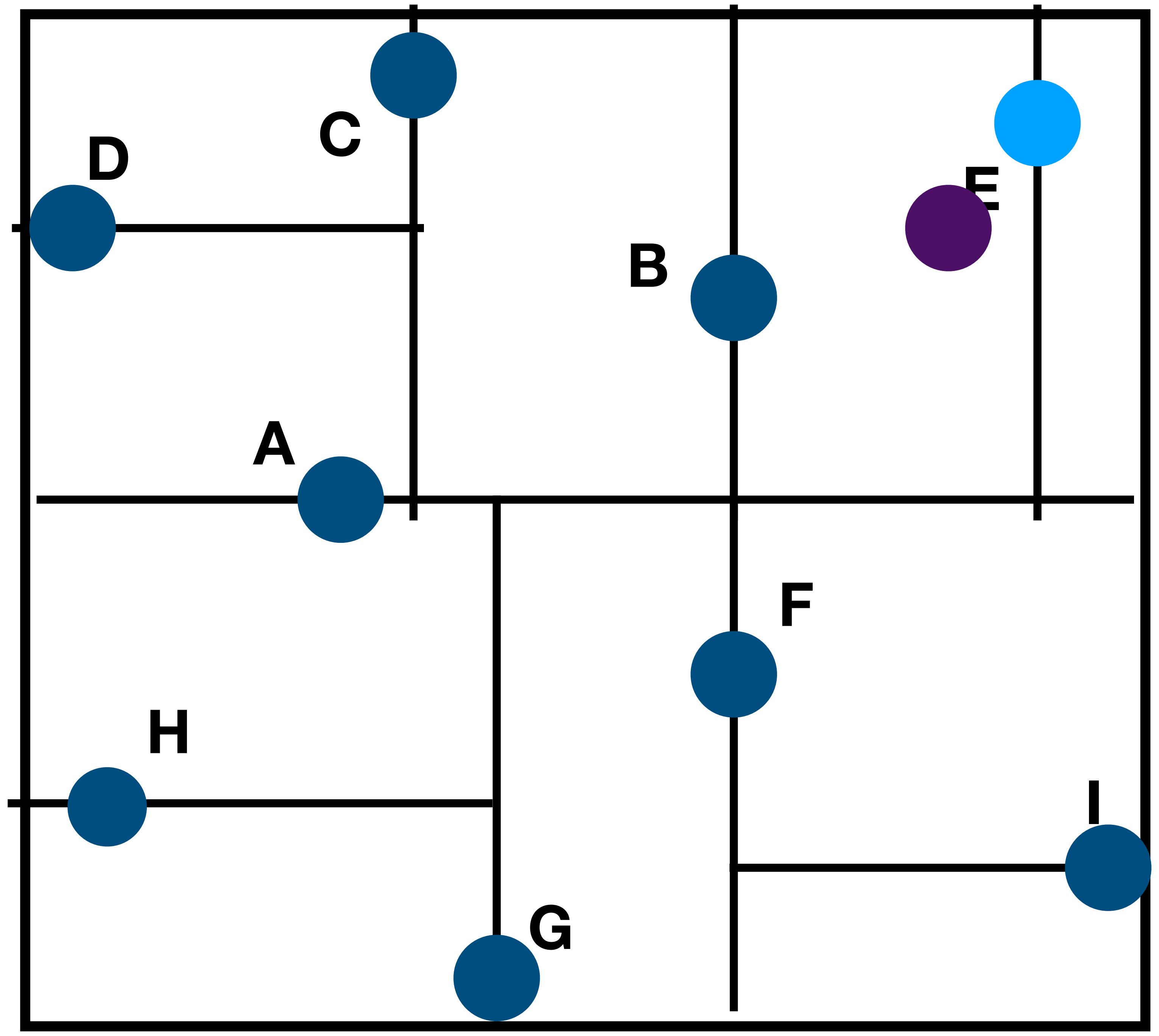
Find closest leaf



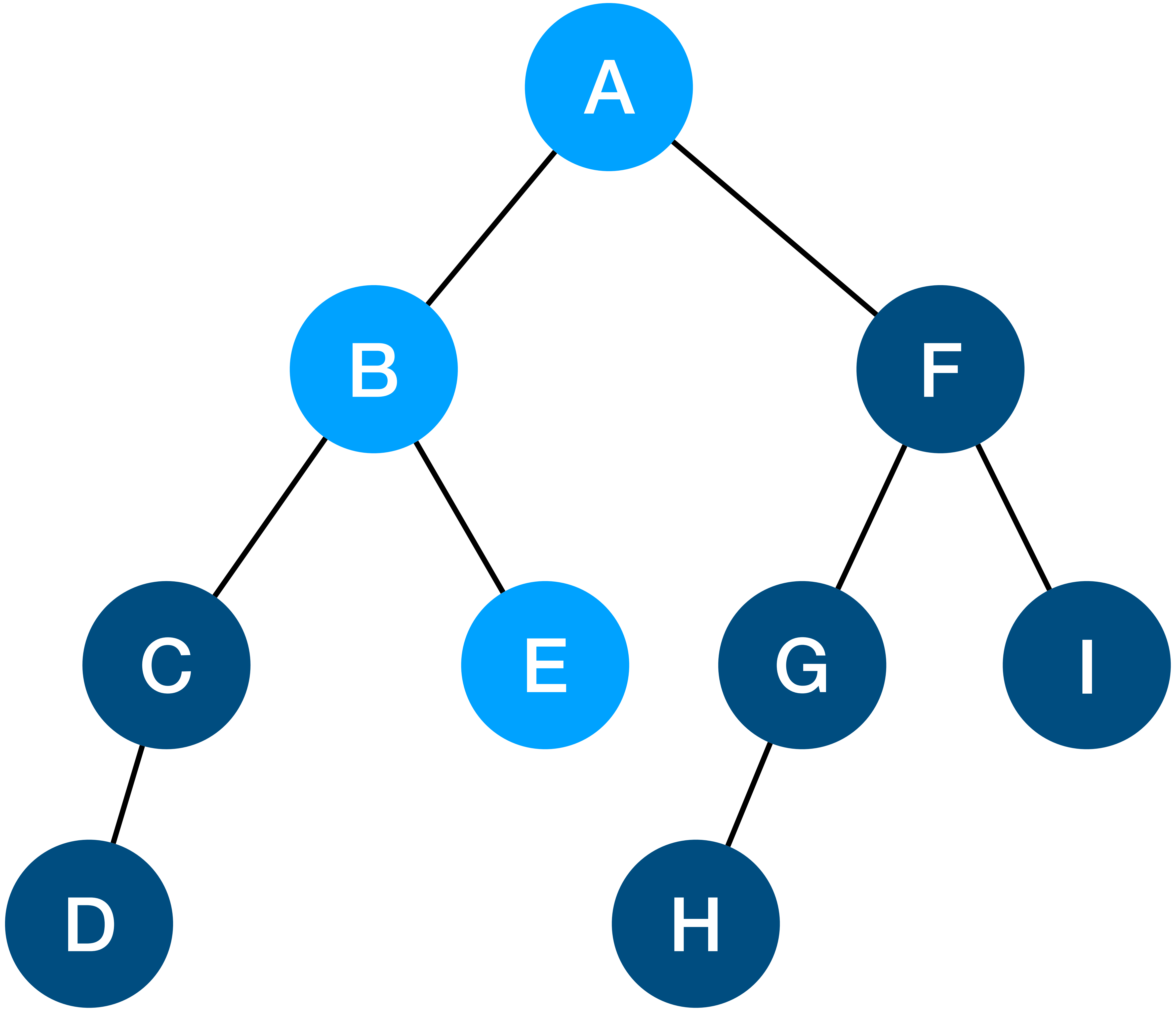


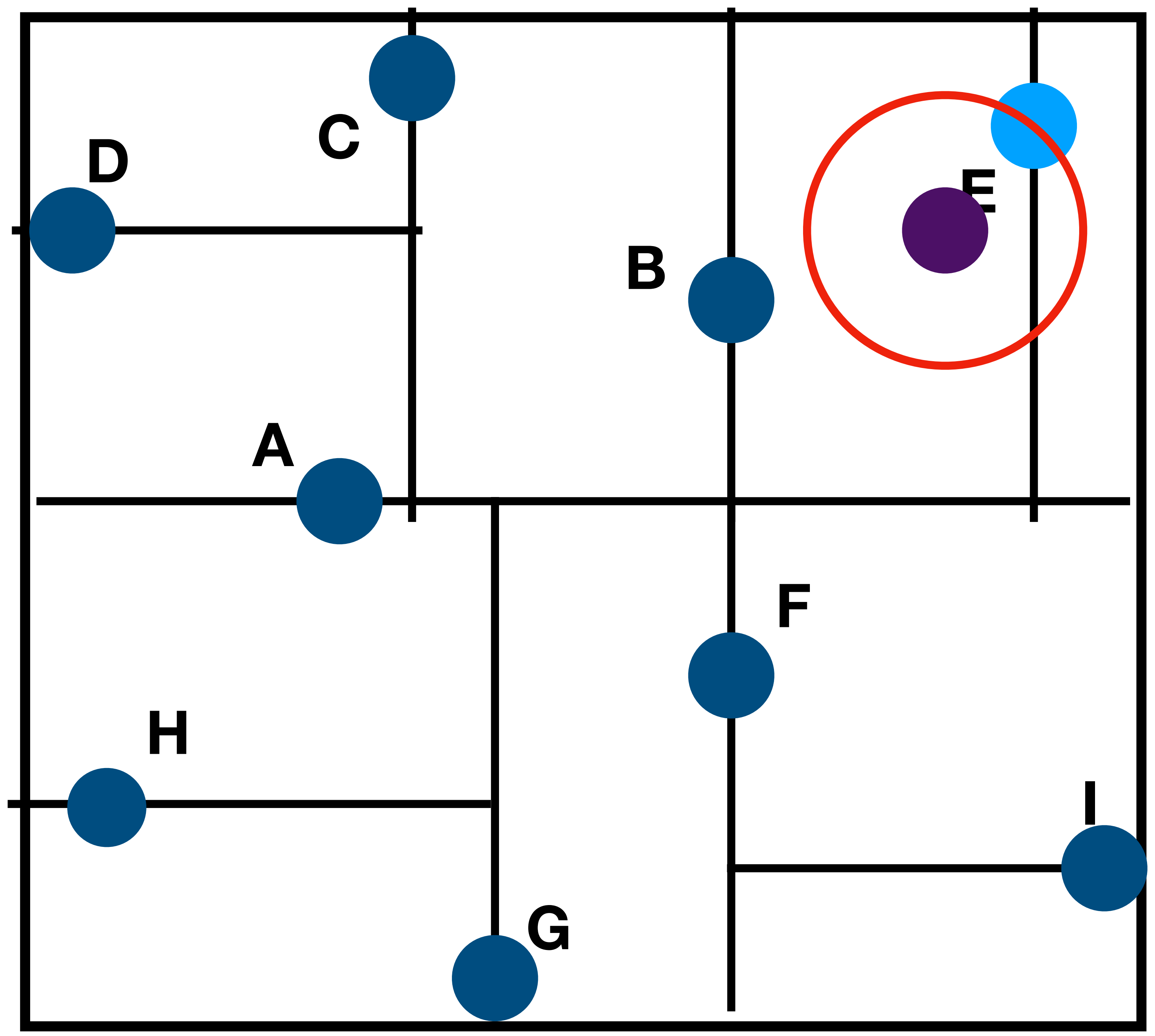
Find closest leaf



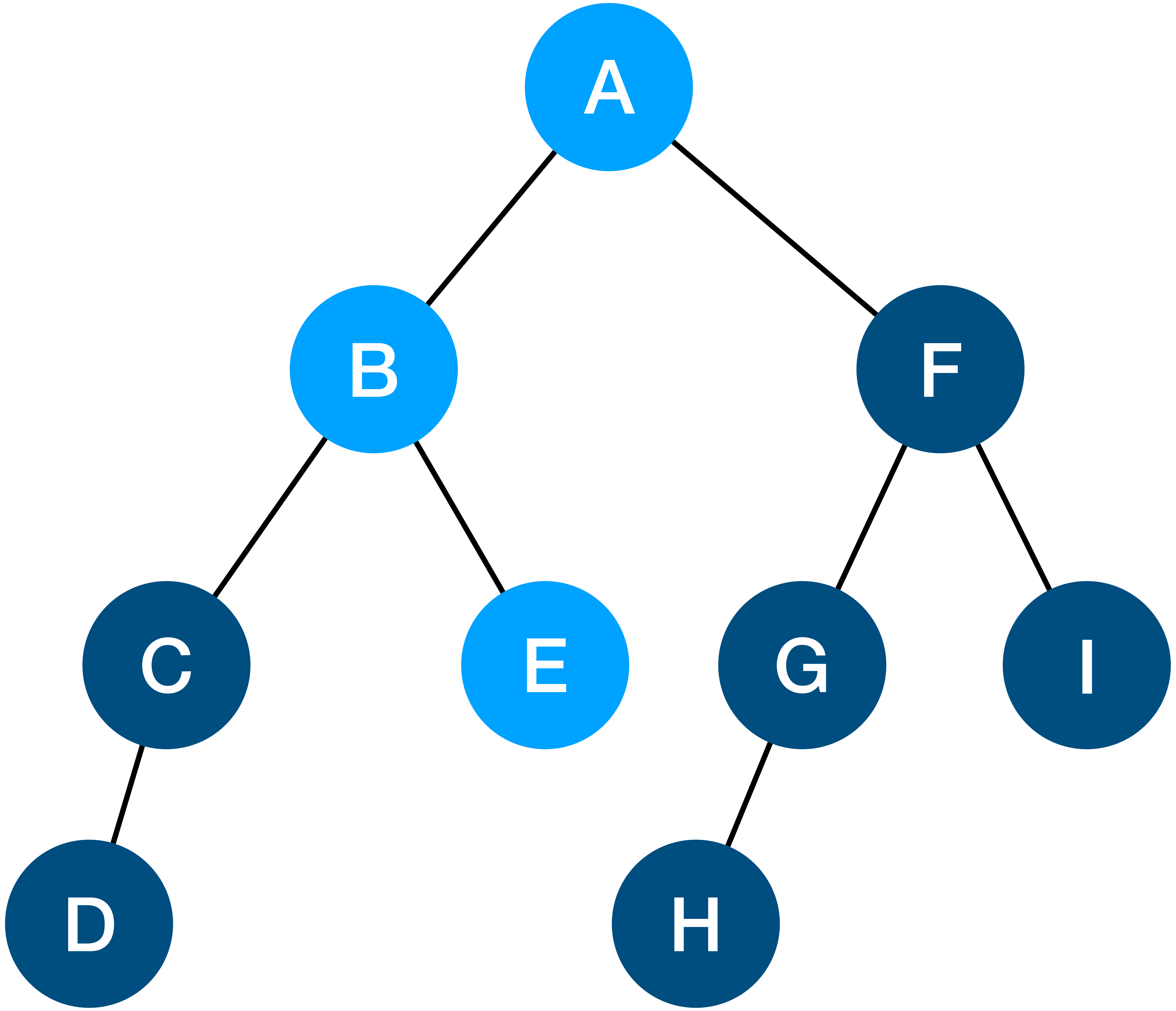


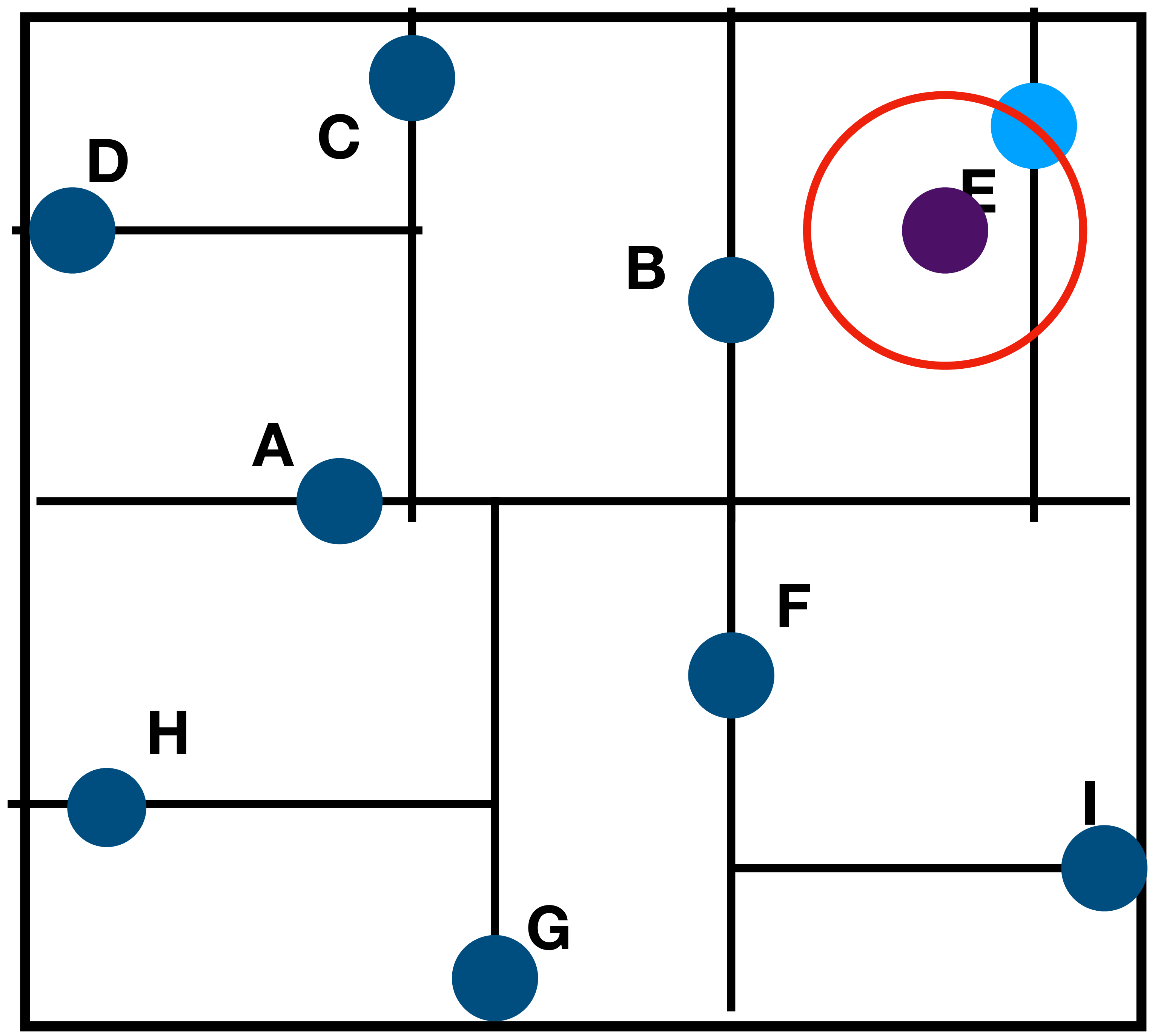
Initial guess for NN



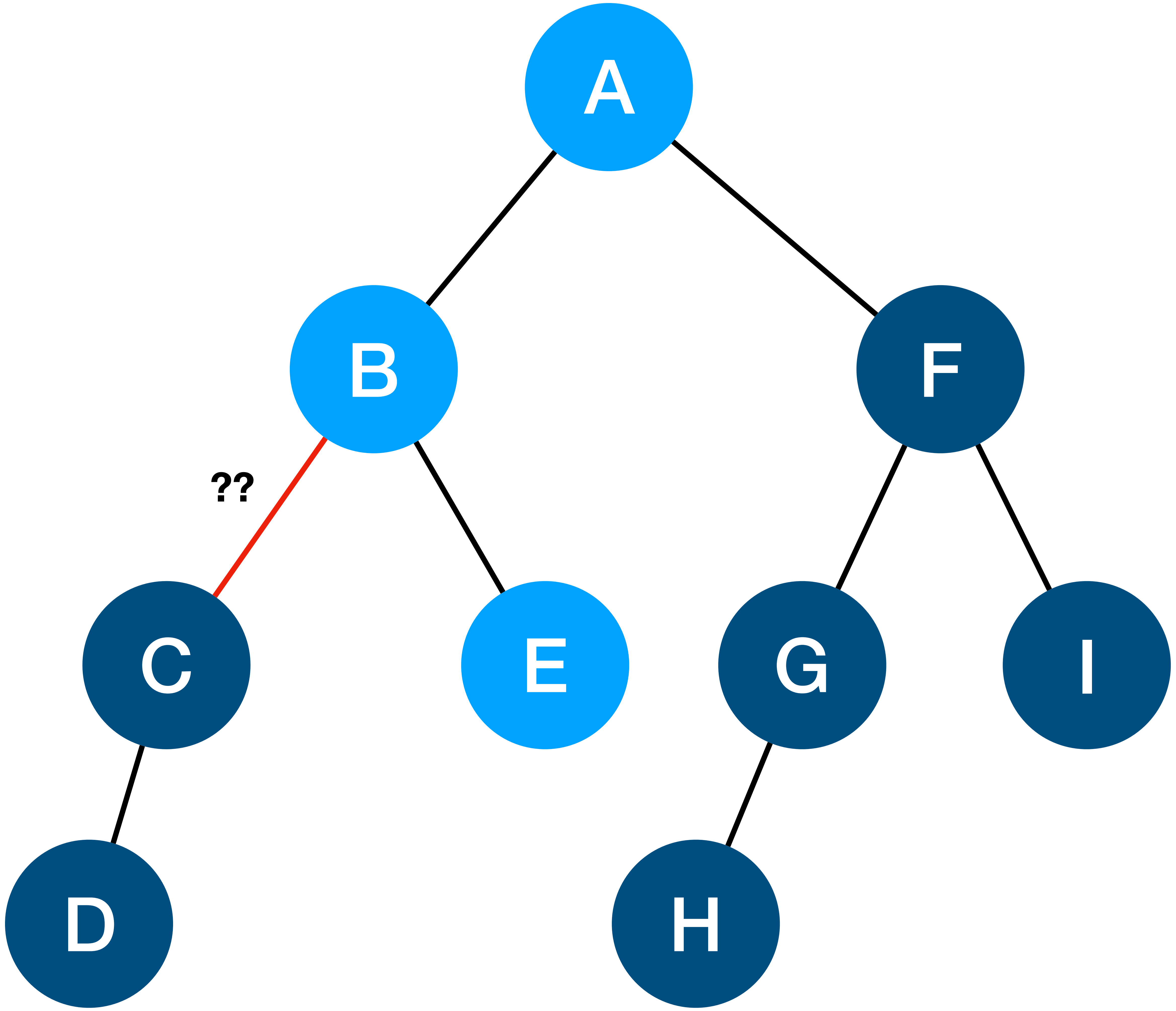


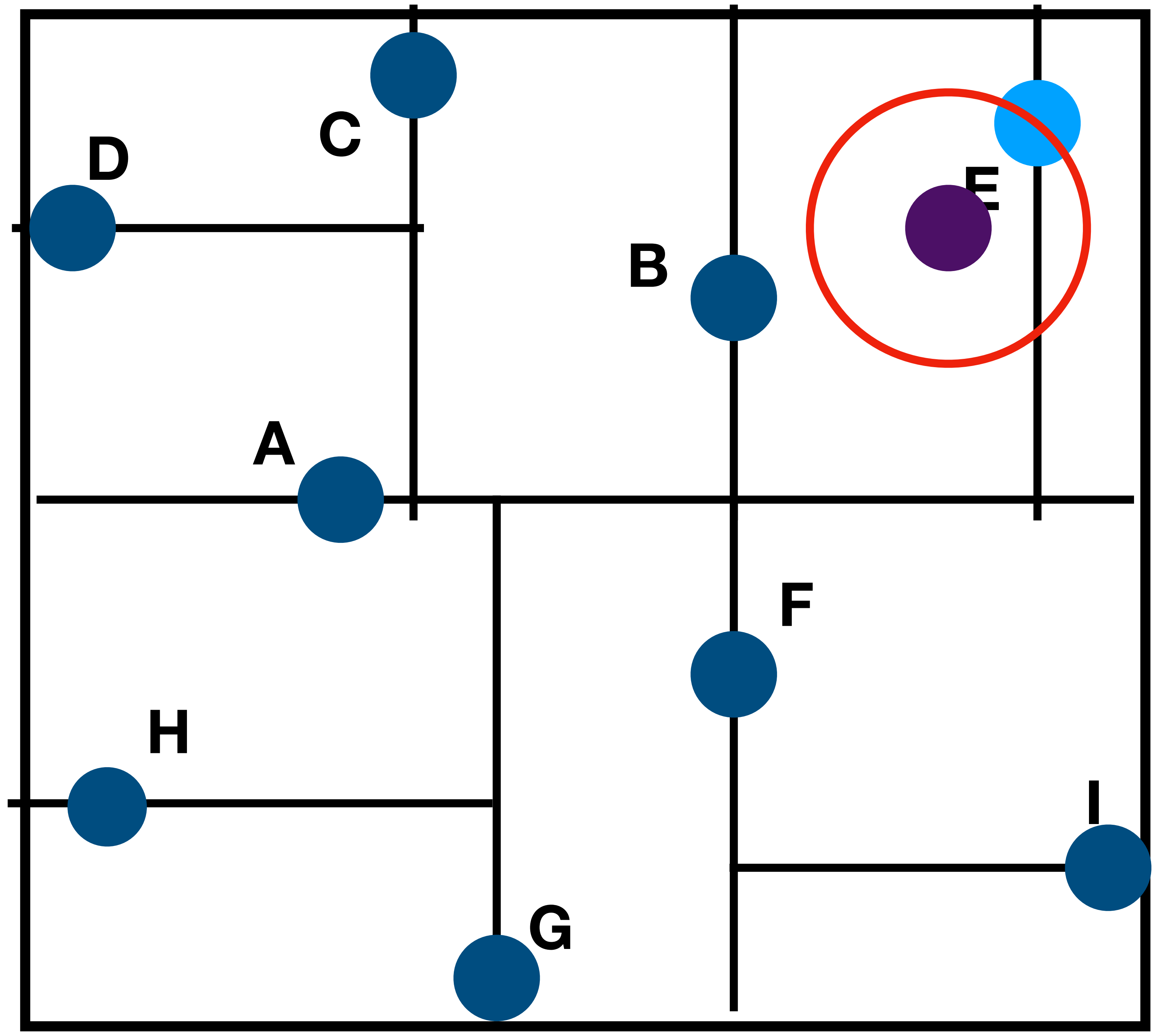
Search rest of tree



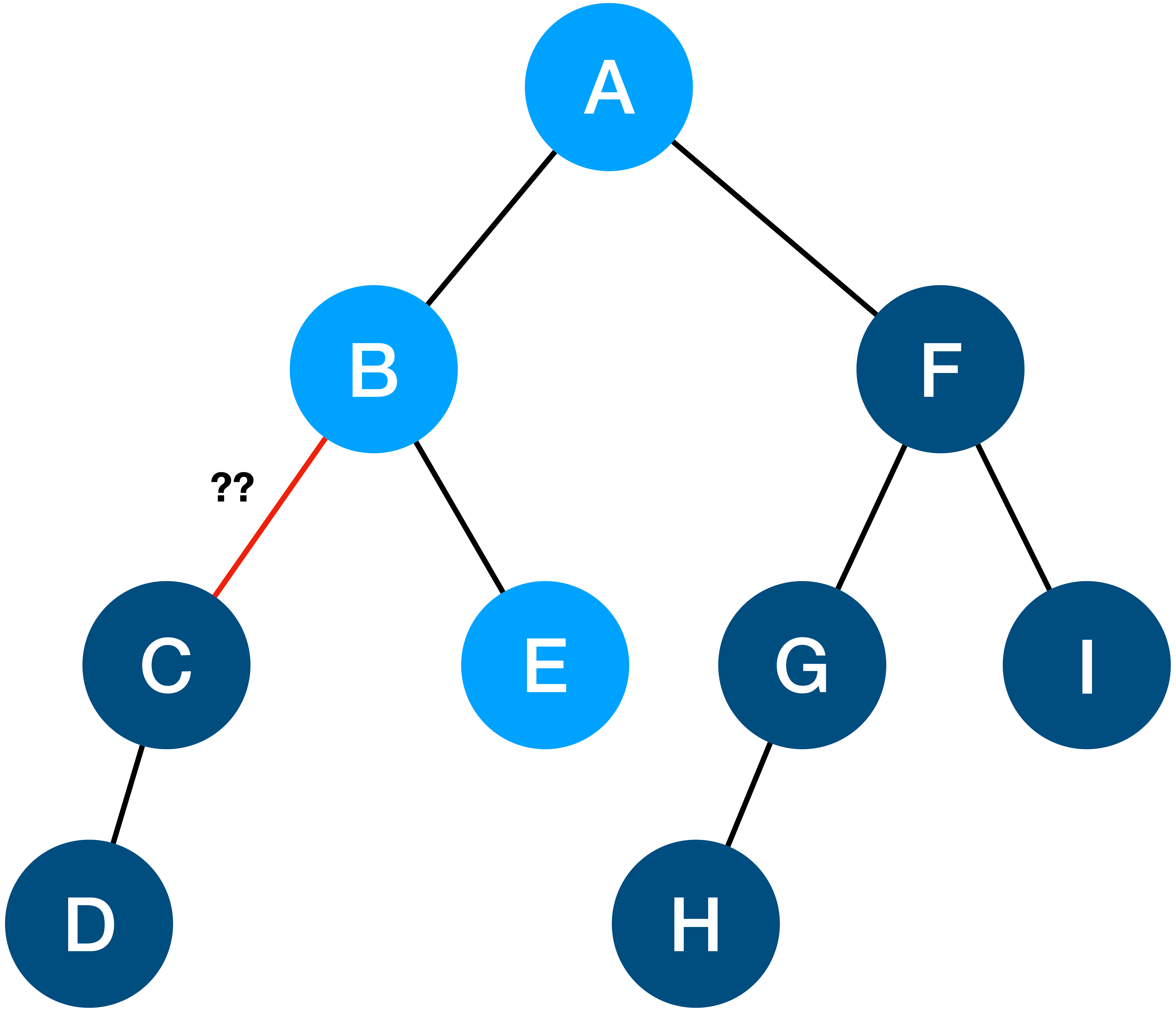


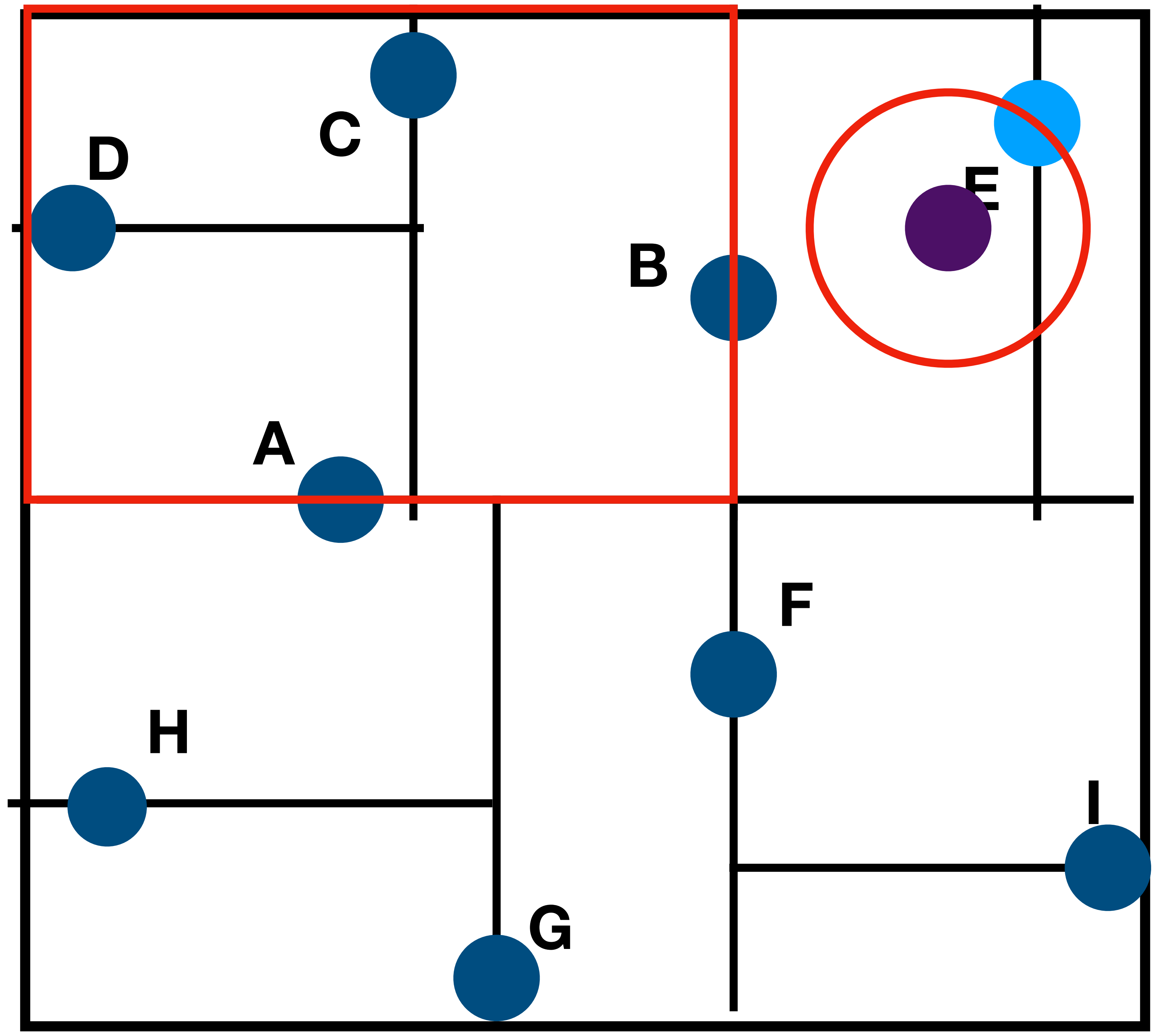
Search rest of tree



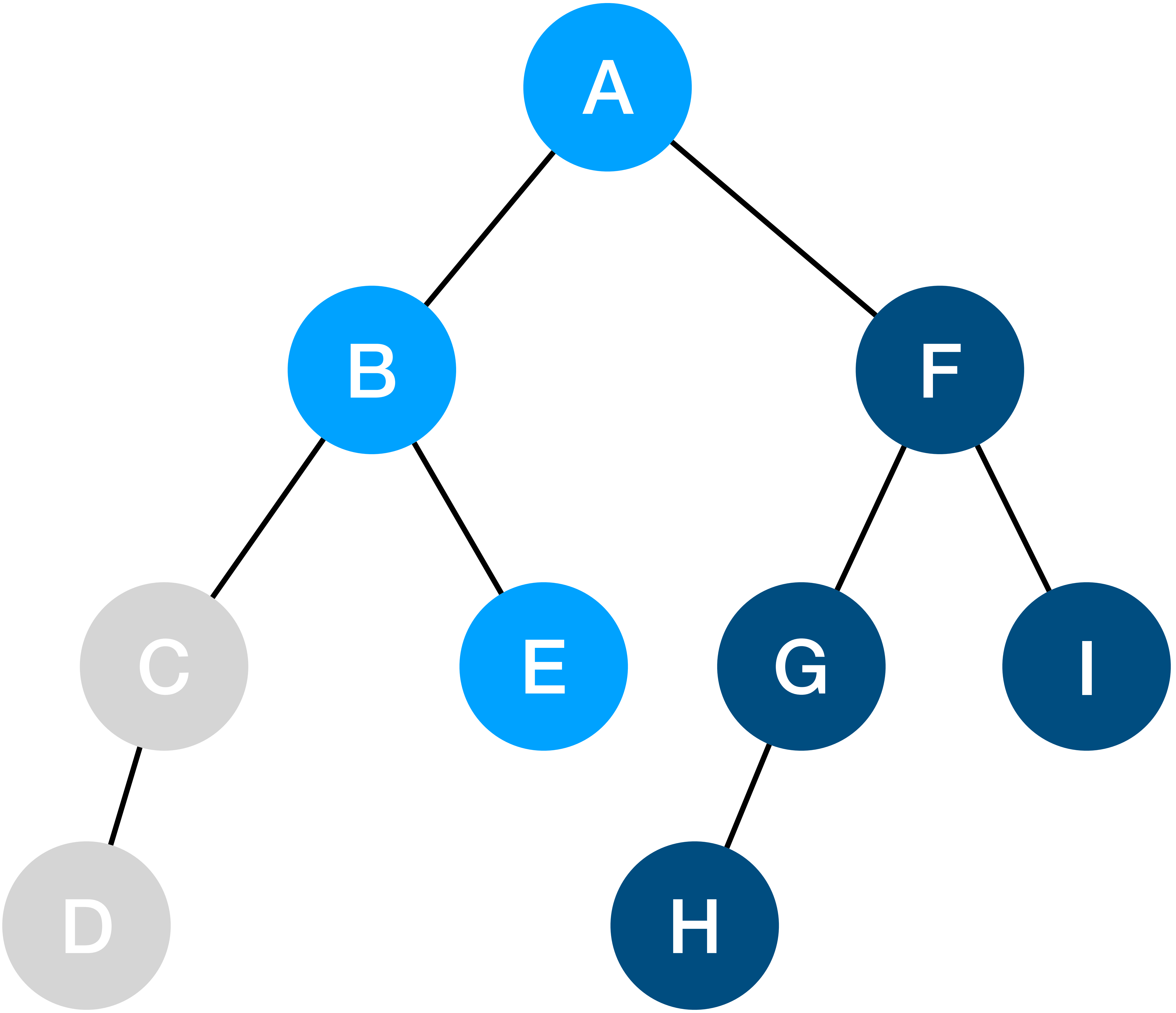


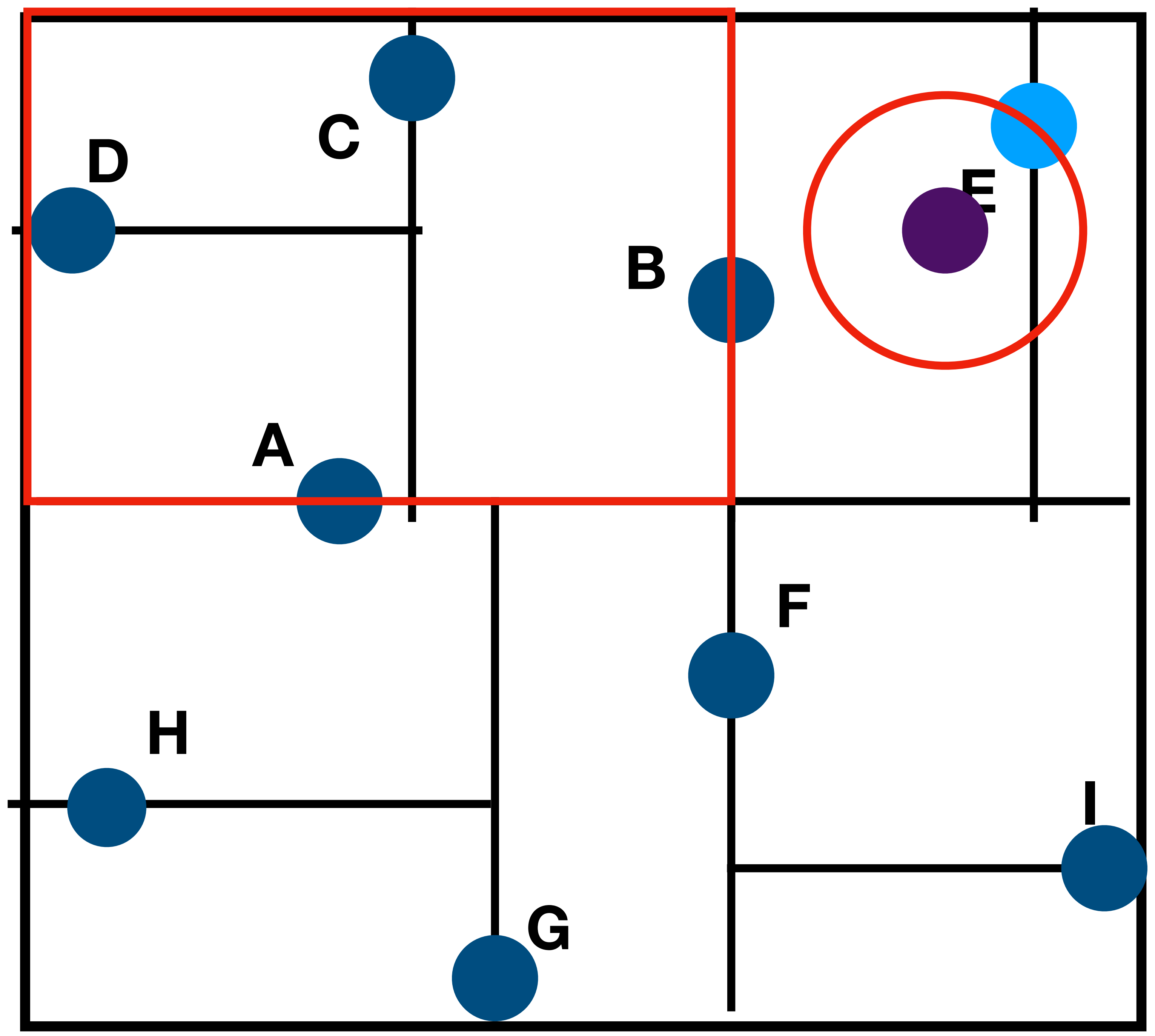
Search rest of tree



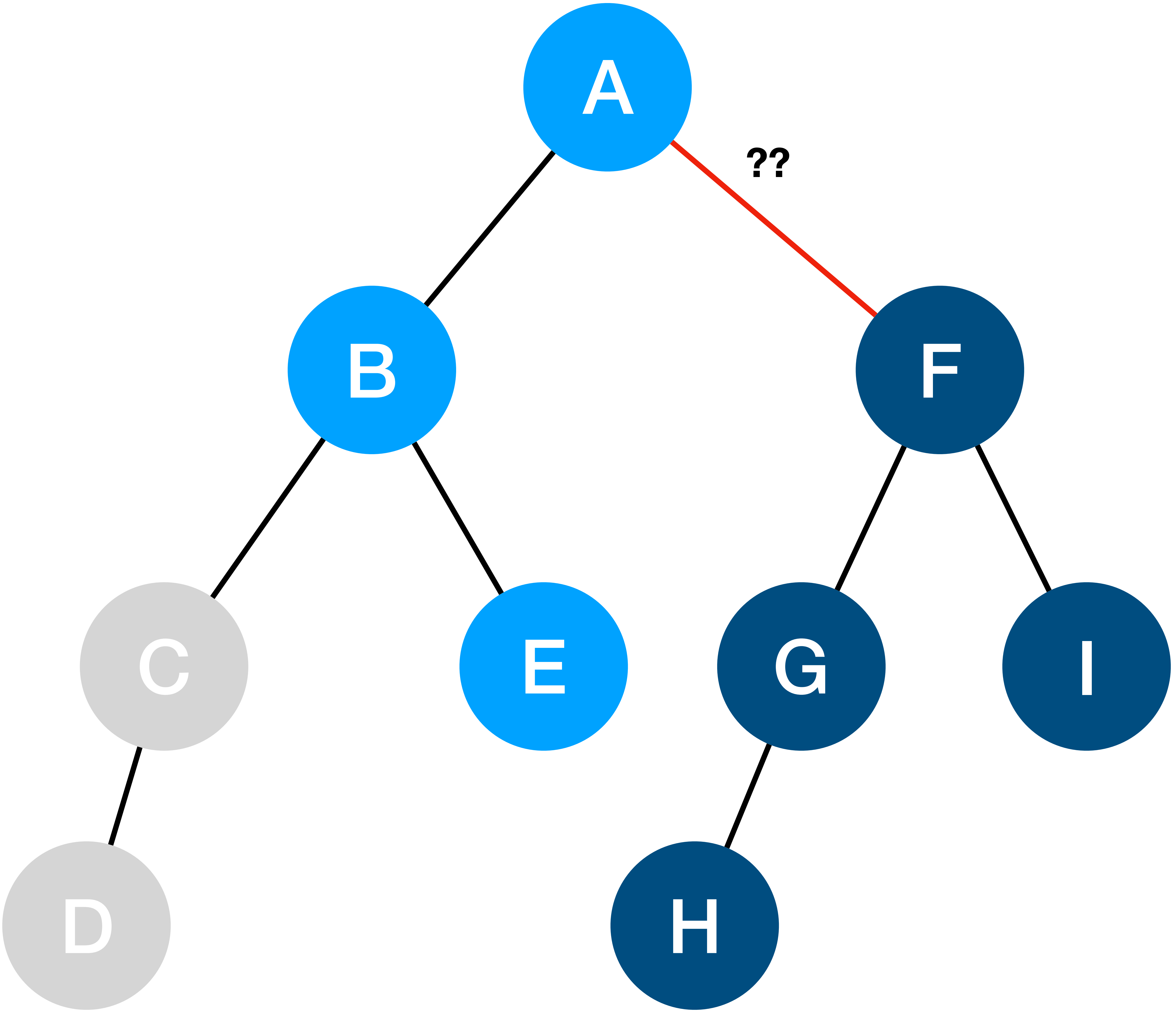


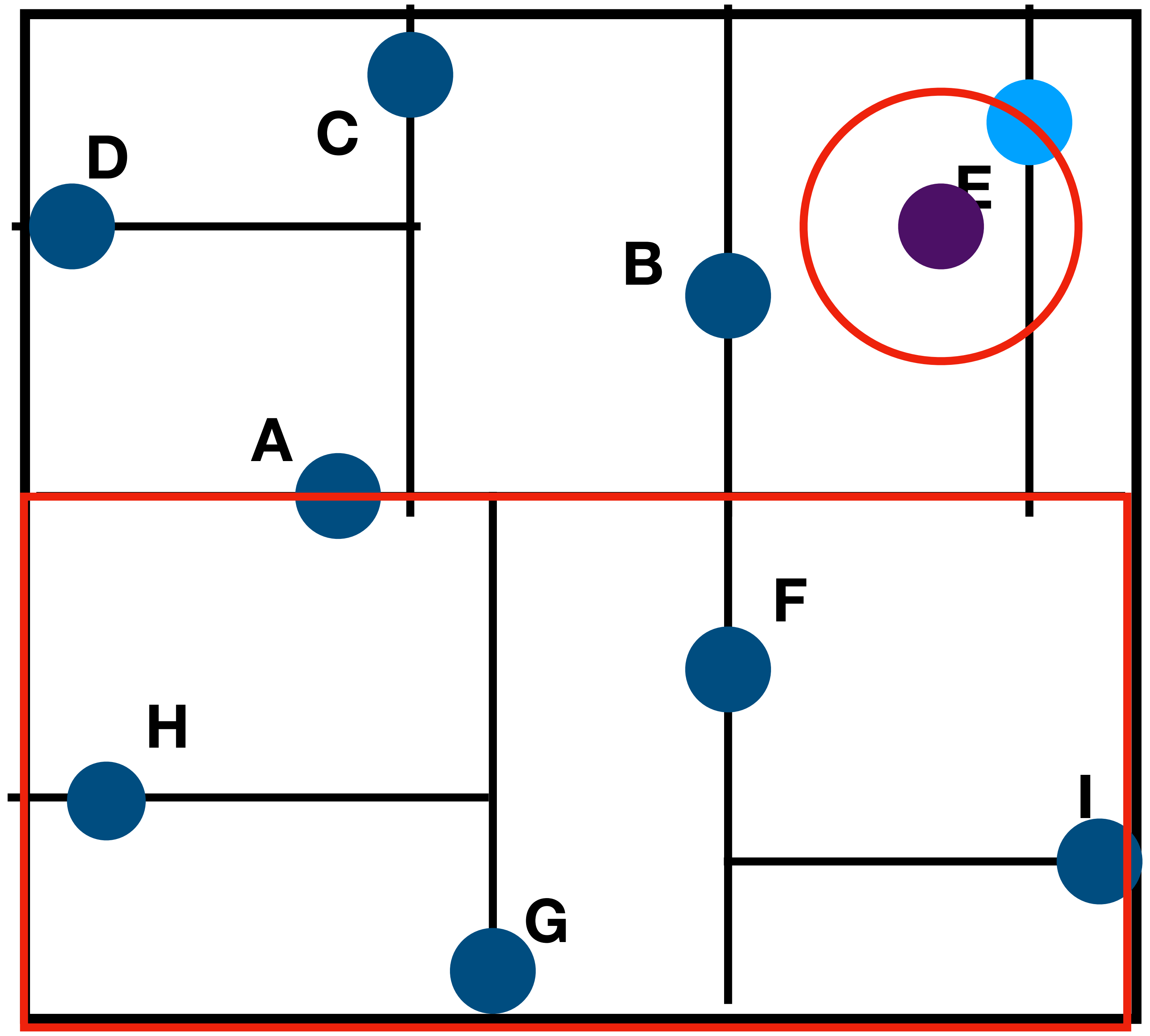
Search rest of tree





Search rest of tree





Search rest of tree