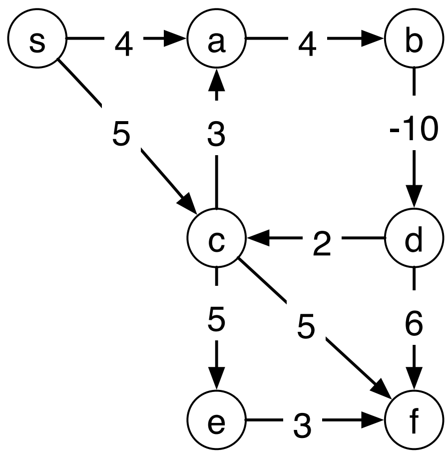
Homework 10

due November 19, 2024

Problem 1:

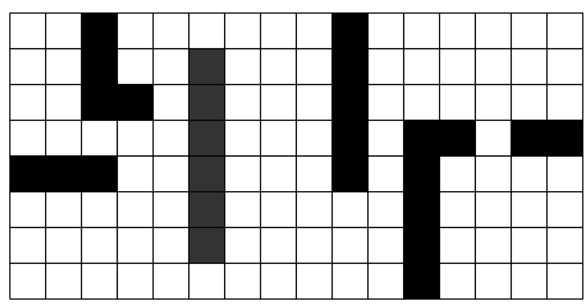
The following graph contains a negative-weight cycle. What happens if we apply Dijkstra's distance algorithm in this case?

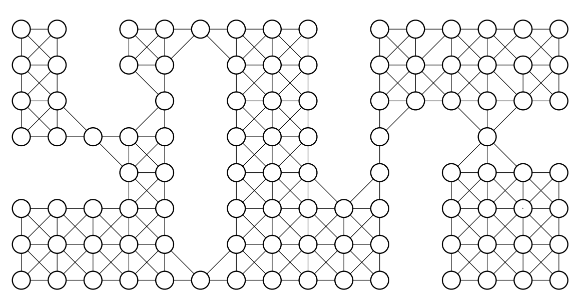
Give the priority queue of vertices after each step.



Problem 2:

Tiled mazes can be represented as graphs, were a tile is adjacent to tiles that they are bordering. There is an example on the right. The weight of the vertical and horizontal edges is one, the diagonal edges are $\sqrt{2}$.





The A* algorithm is a variant of Dijkstra's algorithm used to find a path from a start vertex s to a goal vertex g. It uses a heuristic function h that represents a guess for the distance of a vertex to the goal. In the example of a maze graph on the right, the heuristic function is the Euclidean distance of a vertex to the goal. There, $s = (1,1)$ and $g = (16,8)$, if the start vertex is the upper left corner and the goal vertex is the lower right corner. Thus,

$$(*) \qquad h(x, y) = \sqrt{(16 - x)^2 + (8 - y)^2}.$$

The A* algorithm has two lists, the initially empty *closed lists* and the list of *open list*, consisting initially only of the start node s. We adorn each node v with the current distance from s discovered, called $\delta(v)$. Initially, $\delta(s) = 0$ and $\delta(v) = \infty$ for all other nodes v. We denote $c(a, u)$ as the weight of an edge from Node a to Node u.

While the open list is not empty, do the following:

- Move the node a with the smallest value $\delta(v) + h(v)$ from the open to the closed list.
- If $a = g$, stop.
- For all nodes u adjacent to a:
  - If u is in the closed list, skip u
  - Move u into the open list
  - Determine $\delta(u) = \min(\delta(u), c(a, u) + \delta(a))$
  - If the minimum is the second value, make a the parent of u.

If the open list is empty and we did not terminate earlier, then there is no path.

It can be shown that if h is always equal or smaller than the true costs of a path to g, that this algorithm gives an optimal result:

P. E. Hart, N. J. Nilsson and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," in *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100-107, July 1968. (available online)

Question: Explain why the Euclidean heuristic ($*$) is always equal or smaller than the true cost.

Problem 3:

An undirected graph is called connected if there is a path from every node to every other node. An edge is called a *connector* if the graph after removal of the connector is no longer connected. Recall that in an undirected graph, there are only tree edges and back edges after DFS.

(a) Show that a back edge can never be a connector.

(b) Show that a tree edge (u, v) is a connector if and only if there are no back edges that connect a descendant of v to an ancestor of u.