Problem 1: Traversal

Write breadth-first- and depth-first-search traversals for the graph. Start at node A.

1a) Breadth-First-Search Traversal

$$A, D, B, C, E, G, F$$

1b) Depth-First-Search Traversal

$$A, D, E, B, G, C, F$$

Problem 2: Application

For a 3D puzzle, a 2D matrix represents the length and breadth. Each cell's value gives the height $puzzle[row][column]$. *From cell[0][0], reach cell[m-1][n-1] by moving orthogonally. Write an algorithm to do this with minimal effort. Each route's effort is the maximum absolute difference between two consecutive cells.*

2a) Algorithm

See external file MinPuzzle.py

2b) Time Complexity: $\mathcal{O}(mn * \log(mn))$

The outer while loop iterates mn times, and two inner heap operations $pop()$ and $push()$ are each $\mathcal{O}(log(mn))$.

These operations decide the upper-bound:

Creating the memo matrix and running the while loop is $\mathcal{O}(mn)$, where m and n are the total rows and columns;

In the while loop, $heappop()$ and $heappush()$ are both $\mathcal{O}(log(mn))$.

These operations are constant, $\mathcal{O}(1)$, runtime:

Getting all rows and columns;

Assigning and updating the current cost, row, and column;

Creating the heap;

Assigning and updating the max cost;

Marking each cell visited;

Running the nested loop — i.e., always four times, so O(1);

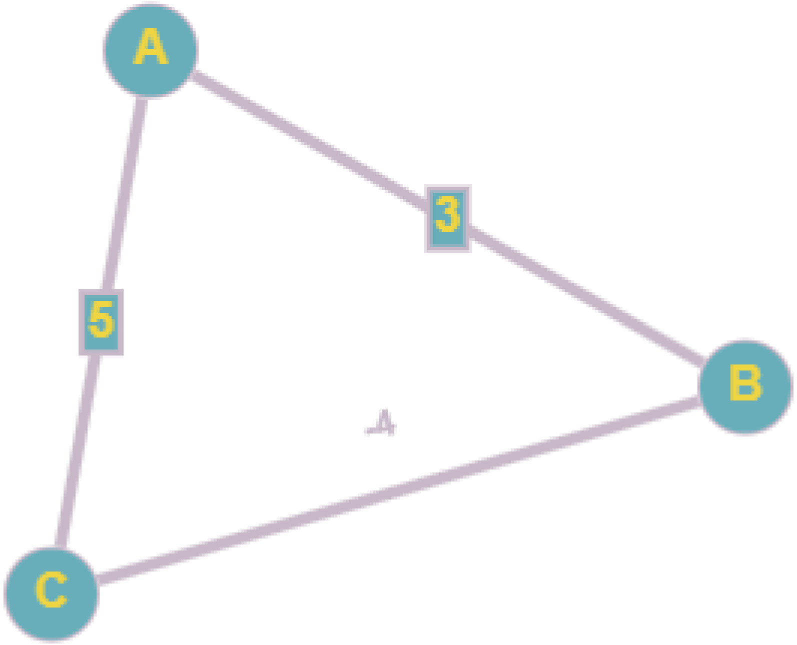
Checking each neighbor with indexing into the array

The complexity can't be $\mathcal{O}(m + n)$, as the while loop doesn't iterate linearly but instead traverses the matrix while considering multiple directions.

Note: This writing resembles my group's submission PDF because I wrote that part of the document.

Problem 3: Dijkstra and Negative Edges

Explain, with a sample graph, why Dijkstra fails with negative edges.



Graph ABC has weights:

$$\begin{aligned} (A,B) &\rightarrow \ \ 3 \\ (A,C) &\rightarrow \ \ 5 \\ (B,C) &\rightarrow -4 \end{aligned}$$

The shortest path for AB should be 3, but it's 1 because ACB is $5 + (-4) = 1$ and $1 < 3$. Dijkstra is a greedy algorithm and only makes the choice in the queue.