2020-04-07 Graphs / Dijkstra's Algorithm

Tuesday, April 7, 2020 7:29 AM

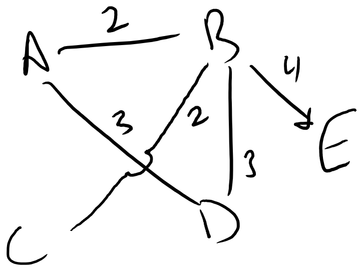
Student Questions

- Some students read out of the wrong OpenDSA book. My fault!
  - https://opendsa-server.cs.vt.edu/ODSA/Books/CS3/html/ is the one with graph information
- When would you choose an adjacency matrix over an adjacency list?

Adjacency Matrix

2D vector

	A	B	C	D	E
A	0	2	-1	3	-1
B	2	0	2	3	4
C	-1	2	0	-1	-1
D	3	3	-1	0	-1
E	-1	4	-1	-1	0



- Advantage:
  - Visual nature of matrix might make it easier to conceptualize the layout of the graph
  - Adjacency matrices tend to be application agnostic (a 2D array is a 2D array)
    - You'll see a lot of graph algorithms use adjacency matrix representations
- Disadvantages
  - Tends to take up more memory when the graph is sparsely populated (not a lot of edges in graph)

Adjacency List / Edge List

- Advantage:
  - Not a lot of wasted space on sparse graphs -- memory is O(E)
- Disadvantages
  - Tend to be application specific

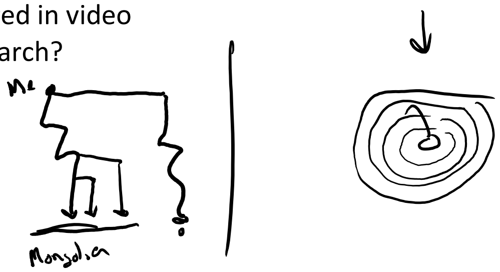
GraphNode	Vertices -- unordered_map<KEY, pair<int, GraphNode>

A	"A": {0, A}
	"B": {2, B}
	"C": {3, C}

B	"A": {2, A}
	"B": {0, B}
	"C": {2, C}
	"D": {3, D}
	"E": {4, E}

- How do you build an adjacency list in code?
  - See code
- How do you add nodes to a graph?
  - See code
- Why doesn't Dijkstra's algorithm work with negative integers?
  - Answered in video

    - See code
- Why doesn't Dijkstra's algorithm work with negative integers?
    - Answered in video
- DFS vs BFS search?


Me
Mongolia