Objectives

* Unravelling inherited classes
* Understanding Graph ADT
* Understanding DSF and BSF Graph Paths

Due Date

This assignment is due on Friday, April 26 by 11:00 pm.

**Remarks:**

* This Homework is a single person effort.

Value

This assignment is worth 20 points.

Activities

Complete HW5 by implementing the circular puzzle in a Graph ADT.

1. Write a function list\_to\_graph(L) that converts the list defining the circular tiles into the Graph ADT definitions (a set of vertices and a set of edges) that can be entered into a Graph ADT. Each tile will be a vertex. The first tile will be tile 1, not tile 0. The weight of each tile provides the definition for two edges. Paste that function under this line.

def list\_to\_graph(L):

vertices = set(range(1, len(L) + 1))

edges = set()

for i in range(len(L)):

weight = L[i]

next\_index = (i + weight) % len(L)

edges.add((i + 1, next\_index + 1, weight))

edges.add((next\_index + 1, i + 1, weight))

return vertices, edges

1. Instantiate a Graph with the circular tile definition created above.
2. Determine if a path exists between the first and last tile.
3. See if you can implement the Graph ADT further to determine the *shortest* path from the first tile to the last tile.
4. See if you can further expand the Graph ADT to determine *all* paths.

**Submit**

1. This Word document with your list\_to\_graph(L) function embedded.
2. A HW11b.py file which contains all the code for your implementation and how you entered each puzzle to find if there is a path.