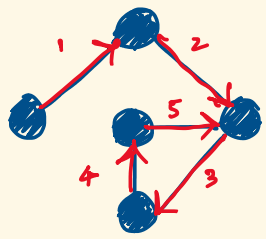
Eulerian tour

Sunday, October 29, 2023 5:36 AM

Eulerian path is a path that visits all edge exactly once.



Eulerian circuit starts & ends at the same circuit.

Finding an Eulerian path in a directed graph

1. Verify that there is a eulerian path.
   ↳ Count in & out degree.
2. Must have exactly 2 nodes with $|in - out| = 1$
3. Every time we visit a node's children, we keep removing the edges.
4. If all edges are removed, add node to path.

```
class EulerianTour:
    def setGraph(self, graph):
        self.adj = defaultdict(list)
        self.indegree = defaultdict(int)
        self.outdegree = defaultdict(int)

        ## calculate indegrees and outdegrees
        for u, v in graph:
            self.indegree[v] += 1
            self.outdegree[u] += 1
            self.adj[u].append(v)

    def isValidGraph(self) -> bool:
        sum_ = 0
        for node in self.adj.keys():
            inDeg, outDeg = self.indegree[node], self.outdegree[node]
            sum_ += (inDeg - outDeg)
        return sum_ == 0

    def getStartNode(self) -> int:
        for node in self.adj.keys():
            inDeg, outDeg = self.indegree[node], self.outdegree[node]
            if abs(inDeg - outDeg) == 1: return node
        return list(self.adj.keys())[0]

    def dfs(self, node, path):
        while len(self.adj[node]):
            nextNode = self.adj[node].pop()
            self.dfs(nextNode, path)
        path.append(node)

    def getEulerianPath(self) -> str:
        ## find the eulerian path
        path = []
        self.dfs(self.getStartNode(), path)
        return ' '.join(map(str, path[::-1]))

eu = EulerianTour()
```

Same algo can be used to find circuit.

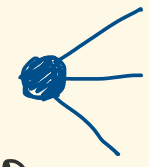
return any

remove the edge

add node when all outgoing edges exhausted.

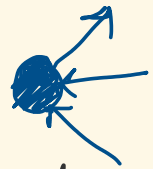
Degree

Undirected



Degree = 3

Directed



Indegree = 2
Outdegree = 1

Prerequisites for a Eulerian path & circuit

	Circuit	Path
Undirected	every node has even degree	Every node has even or exactly 2 nodes with odd degree.
Directed	Every node has eq in & out-degrees	At most one node has in - out = 1 & one node has out - in = 1. Rest all must have same degree

These nodes are start & end nodes of the path

If graph has Eulerian paths, it also has circuit

* If path has more than eq to (edge count + 1), false!