Fleury's Algorithm and Prim's Algorithm Pseudocode

Your Name

October 9, 2025

1 Fleury's Algorithm

Algorithm 1 Fleury's Algorithm for Euler Circuit

Require: Graph *graph* with adjacency matrix *matrix*, start vertex *startVertex*

Ensure: Euler circuit as list of vertices

```
circuit ← empty list
mutableGraph ← graph.copy()
currentVertex ← startVertex
circuit.add(currentVertex)
while hasUnvisitedEdges(mutableGraph) do
    nextVertex ← chooseNextVertex(mutableGraph, currentVertex)
    if nextVertex == −1 then
        break
    end if
    mutableGraph.removeEdge(currentVertex, nextVertex)
    circuit.add(nextVertex)
    currentVertex ← nextVertex
end while
return circuit
```

2 Prim's Algorithm

Algorithm 2 Prim's Algorithm for Connectivity Check

Require: Graph g, start vertex $startVertex$, visited array $visited$

Ensure: Updates visited array to mark connected vertices

```
visited[startVertex] ← true
while true do
    minWeight ← Integer.MAX_VALUE
    nextVertex ← −1
    for i = 0 to g.getSize() − 1 do
        if visited[i] then
            for j = 0 to g.getSize() − 1 do
                if ¬visited[j] and g.hasEdge(i, j) then
                    weight ← g.getWeight(i, j)
                    if weight < minWeight then
                        minWeight ← weight
                        nextVertex ← j
                    end if
                end if
            end for
        end if
    end for
    if nextVertex == −1 then
        break
    end if
    visited[nextVertex] ← true
end while
```