

CSC301 Programming Assignment 2

October 28, 2023

Havin Lim

Pseudocode

```
CollectEverything(graph, start, items, end, comparator, sumComputer):
    fullPath = new List of Integers
    fullPath.add(start)

    itemsList = new HashSet of Integers
    itemsList.addAll(items)

    curr = start

    while itemsList is not empty:
        result = Dijkstra(graph, comparator, sumComputer, curr)
        shortest = null

        for item in itemsList:
            if shortest is null or comparator.compare(shortest.Item1,
result.Get(item).Item1) > 0:
                shortest = result.Get(item)
                curr = item

        itemsList.remove(curr)

        path = DijkstraConstructPath(result, curr)
        first = true

        for i in path:
            if first:
                first = false
                continue
            fullPath.add(i)

        result = Dijkstra(graph, comparator, sumComputer, curr, end)
        append result to fullPath

    return fullPath
```

The method starts by creating an ArrayList fullPath which contains the path that the square moves to reach all the blue rings and the golden ring. Next, we initialize a HashSet itemsList to store the vertexIDs we must visit. Then, the code runs a while loop until the itemsList is empty. In the while loop there also exists a for loop that iterates for all the vertexIDs in the itemsList. It compares the weight of the current item to the stored, current, shortest item. After comparing the current vertexIDs weight to the shortest weight, it keeps the shortest weight in shortest. It constructs a path from the current curr to the selected item and appends the vertices of this path to the fullPath list, skipping the first element. After the loop finishes, the code calculates the path from the last curr (which should be

the last collected item) to the endpoint using `GraphAlgorithms.Dijkstra` and appends this path to the `fullPath` list.