```
func main(Graph G):

    create arr lowest_fluc_cycles   //array of lowest fluctuation cycles for each vertex v

    for vertex v in V: // O(n) iterations

        // find paths to all vertices from v, s.t. paths have lowest fluctuation

        fluc[], prev[] = mod_Dijkstra(G, v)    //O(m + nlogn) call

        create arr candidates  //'shortest' cycles containing v

        for vertex u where there exists e(u,v):     //for all back edges to v
            candidates[u] = (fluc[u] + |e(u,v) - e(u,prev[u])|)   //calculate fluctuation of that cycle

        smallest_fluc_u = minimum(candidates)  //find candidate with smallest fluctuation O(n), quickselect

        lowest_fluc_cycles[v] = (prev[smallest_fluc_u], candidates[smallest_fluc_u]) //store path and fluctuation in function-level array

    solution = minimum(lowest_fluc_cycles)   //select path with lowest fluctuation O(n), quickselect

    return solution[0] //return path of lowest fluctuation

```