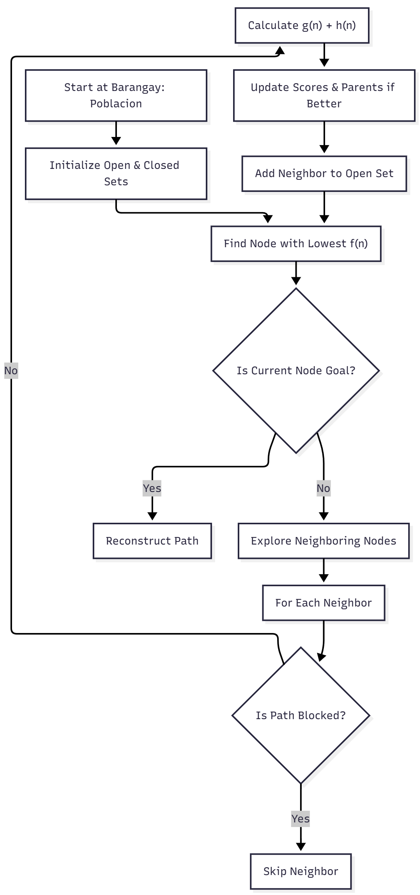


Data Flow Diagram Descriptions

1. External Entities  
   Represent sources or destinations of data outside the system boundary. For this system, external entities could be:
   * The user (emergency responder or local government unit) interacting with the system.
   * External data sources such as geographic data providers or map tile servers.
2. Processes  
   Activities inside the system which transform input data into output data:
   * Data Loading Process: Reads KML files for route geometry and Excel files for travel data.
   * Fuzzy Logic Evaluation Process: Converts slope, travel time, and curvature values into fuzzy costs.
   * *A Pathfinding Process:*\* Calculates optimal routes using fuzzy costs and graph structure.
   * Map Rendering Process: Visualizes barangays, routes, and selected evacuation path on an interactive map.
   * User Interaction Process: Handles user inputs such as barangay selection and displays results.
3. Data Stores  
   Locations where data is stored or retrieved:
   * Route Geometry Store: KML file data holding coordinates of evacuation paths.
   * Travel Data Store: Excel files containing slope and travel times for road segments.
   * Blocked Paths Config: Predefined set of blocked roads stored within system configurations.
4. Data Flows  
   Labeled arrows showing movement of information between external entities, processes, and data stores:
   * User commands triggering data retrieval and route computations.
   * Stream of parsed route data passed to graph construction.
   * Flow of fuzzy cost data feeding the pathfinding algorithm.
   * Output route data to the map rendering module.
   * Metric information moving from algorithms to the dashboard for display.



A\* Algorithm Flowchart

1. Start at Barangay: Poblacion  
   The algorithm begins at the evacuation center barangay named "Poblacion," which is the starting node for pathfinding.
2. Initialize Open & Closed Sets  
   Two collections are initialized:
   * Open set: Nodes that need to be evaluated (starts with Poblacion).
   * Closed set: Nodes already evaluated and for which the shortest path is found.
3. Find Node with Lowest   
   Among nodes in the open set, the algorithm picks the node with the smallest total estimated cost , where:
   * is the actual fuzzy cost from start to node .
   * is the heuristic estimate (straight-line distance) from node  to the goal.
4. Is Current Node Goal?  
   Checks if the chosen node is the destination barangay selected by the user.
   * Yes: Proceeds to reconstruct the path as the optimal route.
   * No: Continues exploring neighbors.
5. Reconstruct Path  
   Backtracks through the parent nodes from the goal to the start node to form the final evacuation route.
6. Explore Neighboring Nodes  
   Examines all barangays directly connected to the current node (per the graph).
7. For Each Neighbor  
   Iterates through each neighbor to process available routes.
8. Is Path Blocked?  
   Determines if the road between the current node and neighbor is on the predefined blocked path list.
   * Yes: Skips evaluating that neighbor to ensure avoidance of blocked roads.
   * No: Proceeds to calculate cost.
9. Calculate   
   Computes:
   * : Updated fuzzy cost from the start node to this neighbor via the current path.
   * : Estimated remaining distance from this neighbor to the goal.
10. Update Scores & Parents if Better  
    If the computed total cost to the neighbor is lower than previous known cost, updates the cost and sets the current node as its parent for path reconstruction.
11. Add Neighbor to Open Set  
    Inserts the neighbor into the open set for future evaluation if not already present.
12. Repeat From Find Node with Lowest   
    The process loops back to selecting the next optimal node to explore until the goal is reached or no path exists.

These descriptions fully explain the internal logic and decision points of the A\* algorithm as used for evacuation pathfinding in the Leon, Iloilo barangays context.