The objective is to develop a system for a local delivery firm with a fleet of three vehicles, each assigned a specific route on a 25x25 grid map that represents a segment of a city. The map delineates sites using coordinates of row numbers and column letters, with the corporate office being at grid square 1A. The paths for each truck are distinctly indicated in various colors (blue, green, yellow), and although all trucks begin on the identical route, each diverges to service certain regions of the city. Black squares denote structures that may necessitate delivery, whilst white squares indicate open areas where vehicles may reroute to access destinations.

Each truck possesses a weight restriction of 1200 kilograms and a maximum volume capacity of 50 cubic meters, accommodating shipments in boxes of 0.5, 1, or 5 cubic meters. Customers provide the weight, volume, and destination address inside a building when submitting shipments. The system must identify the most appropriate vehicle based on proximity and available capacity, considering the truck's capability to divert without traversing buildings. Trucks are designated as full when they attain their maximum capacity in either weight or volume.

**Key Requirements:**

1. **Truck Specifications:**

* Weight Limit: 1200kg
* Volume Limit: 50 cubic meters.
* Package Capacity: 100 packages per truck.

1. **Package Specifications:**

* Weight: Weight of an individual package
* Size: Cubic volume of the package
* ID: A unique identifier of each package

1. **Path Information:**

* Route Points: The path a truck follows, defined as a series of Points (Point structure)
* Path length: Total length of the path each truck

**Structures:**

* **Truck structure:**
  + Contains attributes like Truck Color(m\_truckColor), current load details (currentWeight, currentVolume, n\_package), and a list packages (p[LIMIT\_PACKAGE])
  + Monitors the proportion of weight or volume consumption(p\_truck)
* P**ackage structure:**
  + Includes attributes for weight (p\_weight), size(p\_size), and ID of the package (p\_id)
* **Path structure:**
  + Defines the path using points (po[MAX\_ROUTE]) and tracks the path’s length (p\_length)

**Requirements:**

* **Verify Shipment Information:**
  + Verify that the shipment's weight and dimensions do not surpass the truck's limitations.
  + Confirm that the destination address falls within the grid.
* **Selection of Trucks:**
  + Assess vehicles according to their proximity to the destination and their available capacity (percentage of weight or volume utilized, p\_truck).
  + Utilize the limiting component (weight or volume) as a tiebreaker.
* **Shortest Path Calculation:**
  + Integrate the path length (p\_length) and points (po) into the A\* algorithm.
  + Ensure the algorithm circumvents barriers and identifies the shortest viable path.
* **Manage Full Trucks:**
  + Upon reaching its weight, volume, or package capacity limit (LIMIT\_PACKAGE), the vehicle should be excluded from subsequent deliveries.
  + Store undeliverable parcels for dispatch the next day if all vehicles are at capacity.
* **Delivery Specifications:**
  + Document the chosen vehicle, its itinerary, and any necessary detours to finalize the delivery.

**Functions and Implementation**

* **Calculation of Distance and Path:**
  + Utilize the Path structure for the storage and computation of route data.
  + Utilize the Path structure for the storage and computation of route data.
* **Utilization of Trucks:**
  + Update the Truck structure dynamically when packages are included, recalibrating currentWeight, currentVolume, and n\_package.
* **Package Handling**:
  + Effectively administer packages within each truck with the array (p[LIMIT\_PACKAGE]).
* **Performance Indicators:**
  + Utilize p\_truck (percentage utilization) to assess the appropriateness of trucks during tie-breaking scenarios.