1. (a) Describe and analyze an algorithm to find a legal route with the minimum number of stops from the OOPS warehouse to an arbitrary destination address.

**Solution:** We assume the input graph $G = (V, E)$ is actually a *directed* graph, with two opposing directed edges $u{\to}v$ and $v{\to}u$ for each street segment $uv$, and that each directed edge $e \in E$ stores its direction (north, south, east, or west). We also assume that each vertex has at most one incoming edge with each direction, and at most one outgoing edge with each direction; it follows that $E \le 4V$.

Let $s$ be the start vertex in $G$, and let $z$ and $z'$ be the directed edges in $G$ corresponding to the destination address.

We define a new *directed* graph $G' = (V', E')$ as follows:

- $V'$ contains the start vertex $s$ and every directed edge in $E$. These correspond to interesting locations for the delivery truck.
- $E'$ contains the following edges, corresponding to valid actions by the delivery truck at intersections:
  - $s{\to}(s{\to}u)$ for each directed edge $s{\to}u$ in $G$. (Trucks are allowed to take any street out of the warehouse.)
  - $(u{\to}v){\to}(v{\to}w)$ for every pair of edges $u{\to}v$ and $v{\to}w$ such that
    * $u{\to}v$ and $v{\to}w$ have the same direction (so $u{\to}v{\to}w$ is not a turn), or
    * $u{\to}v{\to}w$ is a right turn at $v$, or
    * $u{\to}v{\to}w$ is a left turn at $v$ and $\deg_G(v) = 2$ (so the turn is forced), or
    * $u = w$ and $\deg_G(v) = 1$ (so $u{\to}v{\to}u$ is a forced U-turn).

  $E'$ contains at most 16 edges of the form $(u{\to}v){\to}(v{\to}w)$ for each vertex $v \in V$, plus at most four edges of the form $s{\to}(s{\to}v)$. Thus, the total number of edges in $E'$ is at most $16V + 4 = O(V)$.

Every walk in $G$ through $k$ intersections corresponds to a walk in $G'$ with length $k + 1$; for example, the walk $s{\to}t{\to}u{\to}v{\to}w$ in $G$, which passes through three intersections $(t, u, v)$, corresponds to the walk $s{\to}(s{\to}t){\to}(t{\to}u){\to}(u{\to}v){\to}(v{\to}w)$ in $G'$, which has length 4.

We compute the **shortest paths** in $G'$ from $s$ to both $z$ and $z'$ using **breadth-first search** starting at $s$, and then return the length of the shorter of these two paths minus 1.

The algorithm runs in $O(V' + E') = \boldsymbol{O(V)}$ ***time***. ■

**Rubric:** 5 points: standard graph reduction rubric (scaled, with implicit brute-force graph construction).

- This is not the only way to describe this graph $G'$. For example, each vertex of $G'$ could specify a vertex of $G$ and an arrival direction, or two vertices of $G$.
- No penalty for returning the length of the optimal route instead of the route itself.
- −½ for off-by-one error in the reported route length.
- −½ for reporting running time as "$O(V + E)$".

(b) Describe and analyze an algorithm to find a legal route with the minimum number of stops, from the OOPS warehouse, to an arbitrary destination address, and then back to the warehouse.

**Solution:** We make the same assumptions as in part (a). Let $s$ be the start vertex in $G$, and let $z$ and $z'$ be the directed destination edges in $G$.

We construct the same graph $G'$ as our solution to part (a), but with (up to four) additional edges $(v{\to}s){\to}s$ for every edge $v{\to}s$ in the directed input graph $G$. Let $dist(x,y)$ denote the shortest-path distance from $x$ to $y$ in $G'$. We compute four ***shortest-path*** distances using ***breadth-first search***:

- $dist(s,z)$ using BFS starting at $s$,
- $dist(s,z')$ using BFS starting at $s$,
- $dist(z,s)$ using BFS starting at $z$,
- $dist(z',s)$ using BFS starting at $z'$.

Finally, we return

$$\min\left\{dist(s,z)+dist(z,s),\ dist(s,z')+dist(z',s)\right\}-1$$

Each invocation of breadth-first search runs in $O(V'+E')=O(V)$ time, so the entire algorithm runs in ***$O(V)$ time***. ■

**Rubric:** 5 points: standard graph reduction rubric (scaled, with implicit brute-force graph construction).

- This is not the only correct algorithm. For example, we could construct a similar graph $G''$ that models forbidding right turns (either by brute force or by reversing every edge in $G'$), and then return

$$\min\left\{dist_{G'}(s,z)+dist_{G''}(s,z'),\ dist_{G'}(s,z)+dist_{G''}(s,z')\right\}-1$$

- No penalty for returning the length of the optimal route instead of the route itself.
- −½ for off-by-one error in the reported route length.
- −½ for reporting running time as “$O(V+E)$”.

2. Describe and analyze an algorithm to find the length of the shortest awesome walk from $s$ to $t$.

**Solution:** Let $G = (V, E)$ be the input graph. We assume every edge in $E$ stores its color. We also assume without loss of generality that $G$ is connected, since any walk in $G$ lies within a single component, and therefore $V \leq E + 1$. Finally, we assume that $s \neq t$, since otherwise the correct output is trivially 0.

We construct a new *directed* graph $G' = (V', E')$ as follows:

- $V' = \{s\} \cup V \times \{red, green, blue\} \times \{1, 2\}$ — Each vertex $(v, c, k)$ indicates that we are located $v$ and we just traversed $k$ edges in a row with color $c$.
- $E'$ contains the following directed edges, for every color $c \in \{red, green, blue\}$:
  - $s \to (v, c, 1)$ for every edge $sv$ with color $c$.
  - $(u, c, 1) \to (v, c, 2)$ for every edge $uv$ with color $c$.
  - $(u, c', 1) \to (v, c, 1)$ for every color $c \neq c'$ and every edge $uv$ with color $c$.
  - $(u, c', 2) \to (v, c, 1)$ for every color $c \neq c'$ and every edge $uv$ with color $c$.

  Altogether, $G'$ has $6V + 1 = O(V)$ vertices and $\deg_G(s) + 5E = O(E)$ edges.

We need to compute the ***shortest path*** in $G'$ from $s$ to any vertex of the form $(t, c, k)$. We compute all such shortest paths using a single ***breadth-first search*** starting at $s$, and then return the minimum of those six distances. The algorithm runs in $O(V' + E') = \boldsymbol{O(E)}$ ***time***. ■

**Rubric:** 10 points: standard graph reduction rubric (with implicit graph construction)

3. Describe and analyze an algorithm to answer the ethnographers' problem. Your algorithm should either output possible dates of birth and death that are consistent with all the stated facts, or it should report correctly that no such dates exist.

**Solution:** This question is asking to find an *ordering* satisfying certain constraints, so we should immediately guess that the solution will involve a topological sort. The only problem is to define a directed graph that properly encodes the given constraints.

Suppose we are given $m$ facts about a total of $n$ distinct people. Recall that each person is identified with a unique integer between 1 and $n$. We define a directed graph $G$ as follows:

- $G$ has $2n$ vertices $b_1, b_2, \dots, b_n, d_1, d_2, \dots, d_n$. For each index $i$, node $b_i$ represents person $i$'s birth date, and $d_i$ represents person $i$'s death date.
- Every directed edge in the graph goes from an earlier event to a later event; thus, the arrow $\rightarrow$ can be read "happened before". There are three classes of edges in $G$:
  - $G$ contains the edge $b_i \rightarrow d_i$ for each $i$, encoding the fact that person $i$ was born before they died.
  - For each constraint of the form "$i$ died before $j$ was born", $G$ contains the edge $d_i \rightarrow b_j$.
  - For each constraint of the form "$i$ and $j$ were alive at the same time", $G$ contains the edges $b_i \rightarrow d_j$ and $b_j \rightarrow d_i$.

  Altogether, $G$ has at most $n + 2m$ edges.

If $G$ is acyclic, then any ***topological ordering*** of the birth and death dates satisfies all the given constraints. The precise dates don't matter; for example, we could declare the $k$th event in some topological ordering to occur exactly $k$ fortnights after Sunday, March 18, 3952BCE.[a] On the other hand, if $G$ has a directed cycle, the given constraints cannot be satisfied.

We construct $G$ in $O(n+m)$ time from the given list of facts by brute force. We then ***topologically sort*** $G$ in $O(V+E) = O(m+n)$ time, after which we can assign suitable dates in $O(n)$ additional time. The entire algorithm runs in $\boldsymbol{O(n+m)}$ ***time***. ■

[a]This is the date of the creation of the universe, according to the Venerable Bede's seminal treatise *De temporibus [On Times]*, published in 703CE. Bede's calculations were considered heretical at the time, because they relied on Hebrew scriptures rather than the more recent Greek translation (the Septuagint) preferred by the Roman Church, which implied a creation date roughly 1700 years earlier.

**Rubric:** 10 points, standard graph reduction rubric (with implicit brute-force graph construction). This is more detail than necessary for full credit. +1 extra credit for properly citing the Venerable Bede instead of James Ussher.

- −1 for reporting the final running time as $O(V+E)$ instead of $O(n+m)$. The input is a list of facts, not a graph or a pair of graphs.