

```
pub struct RoutingData {  
    // relevant nodes and their position  
    pub osm_nodes: HashMap<i64, OsmNode>,  
    // [n_id] -> osm_n_id  
    pub internal_nodes: Vec<i64>,  
    // [e_id] -> target_n_id|length|constraints  
    pub internal_edges: Vec<RoutingEdge>,  
    // [n_id] -> e_id  
    pub internal_offset: Vec<usize>,  
    // tmc_loc -> set<internal_edge_id>  
    pub tmc_mapping: HashMap<u32, HashSet<usize>>,  
    // tmc_loc -> tmc_loc  
    pub tmc_next: HashMap<(u32, bool), u32>,  
}
```

```
pub struct State {  
    pub data: RoutingData,  
    pub grid: Grid,  
}
```

```
pub struct Grid {  
    pub bbox: BoundingBox,  
    pub bin_count_lat: usize,  
    pub bin_count_lon: usize,  
    pub bins: Vec<Bin>,  
}
```

```
pub struct OsmNode {  
    pub position: Position,  
    pub internal_id: usize  
}
```

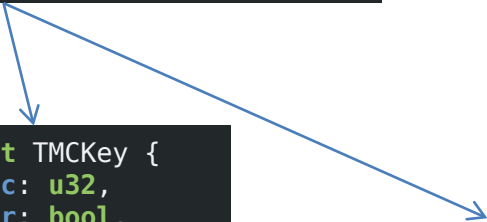
```
pub struct RoutingEdge {  
    pub source: usize,  
    pub target: usize,  
    pub length: f64,  
    pub speed: f64,  
    pub constraints: u8,  
}
```

```
pub struct Position {  
    pub lat: f64,  
    pub lon: f64,  
}
```

```
pub struct Bin {  
    pub nodes: Vec<i64>  
}
```

```
pub struct BoundingBox {  
    pub min_lat: f64,  
    pub min_lon: f64,  
    pub max_lat: f64,  
    pub max_lon: f64  
}
```

```
pub struct TMCState {  
    pub edge_events: HashMap<usize, f64>,  
    pub tmc_events: HashMap<TMCKey, TMCEvent>  
}
```



```
pub struct TMCKey {  
    pub loc: u32,  
    pub dir: bool,  
    pub event: u32,  
}
```

```
pub struct TMCRawEvent {  
    pub loc: u32,  
    pub dir: bool,  
    pub event: u32,  
    pub ext: u32  
}
```

```
pub struct TMCEvent {  
    pub desc: String,  
    pub slowdown: f64,  
    pub ext: u32,  
    pub timeout: u32,  
    pub edges: HashSet<usize>  
}
```