Algorithm 1 Calculate best paths for buses

```
Input: prisioners, buses, graph
Output: paths of all buses

$results \leftarrow [\ ]$
for all prisoners do
  if !findBusType($prisoner$) then
    setType($prisoner$, 'Any')
  end if
end for

for all buses do
  $destinations \leftarrow [\ ]$
  for all prisoners do
    if hasPath($bus$,$prisoner$) then
      if checkType($bus$,$prisoner$) then
        add($destinations$, $prisoner$)
      end if
    else
      erase($prisoner$)
    end if
  end for
  $bus.destinations \leftarrow destinations$
end for

while any bus destinations not empty do
  $ignoredVertexes \leftarrow [\ ]$
  for all buses do
    if !empty($bus.destinations$) then
      $hadAction \leftarrow$ false
      dijkstra($bus.location$)
      $nextDest \leftarrow$ pop($bus.destinations$)
      $pathToNext \leftarrow$ getPath($bus.location$, $nextDest$)
      $currentLocation \leftarrow bus.location$
      $bus.location \leftarrow nextDest$
      prisonerLoop:
      for all prisoners do
        if $prisioner.weight > bus.capacity$ then
          go to $prisonerLoop$
        end if
        if $prisioner.start = bus.location$ and !pickedUp($prisoner$) and checkType($bus$,$prisoner$) then
          $currentNextDist \leftarrow$ dist($nextDest$)
          for all buses as $bus2$ do
            djikstra($bus2.location$)
            $nextDist \leftarrow$ dist($nextDest$)
            $ignoreDest \leftarrow$ false
            for all prisoners as $prisoner1$ do
              if $bus$!=$bus2$ and pickedUp($prisoner1$) and !delivered($prisoner1$) and getBus($prisoner1$)=$bus2$ and checkType($bus2$,$prisoner1$) then
                dijkstra(destination($prisoner1$))
                if dist($nextDest$)< $nextDist$ then
                  add($ignoredVertexes$, $nextDest$)
                  $ignoreDest \leftarrow$ true
                  break loop
                end if
              end if
            end for
            **if** $ignoreDest$ **then**
              **break loop**
            **end if**
            **if** $bus != bus2$ and $currentNextDist > nextDist$ and checkType($bus2$,$prisoner$) **then**
              add($ignoredVertexes$,$nextDest$)
              **break loop**
            **else if** $bus2$ = lastBus **then**
              **if** canFit($bus$, $prisoner$) **then**
                pickUp($prisoner$, $bus$)
                add($bus.destinations$, destination($prisoner$))
              **end if**
            **end if**
          **end for**
        **else if** destination($prisoner$)=$bus.location$ and pickedUp($prisoner$) and getBus($prisoner$)=$bus$ **then**
          deliver($prisoner$, $bus$)
          $hadAction \leftarrow$ true
        **end if**
      **end for**
      **if** $hadAction$ or empty($bus.destinations$) **then**
        **if** $!hadAction$ **then**
          $bus.location \leftarrow currentLocation$
        **end if**
        **for all** $ignoredVertexes$ **do**
          addDestination($bus$, $ignoredVertex$)
        **end for**
        $ignoredVertexes \leftarrow [\,]$
      **else**
        $bus.location \leftarrow currentLocation$
        **repeat loop iteration**
      **end if**
      **if** $hadAction$ **then**
        $results[bus] \leftarrow pathToNext$
      **end if**
    **end if**
  **end for**
**end while**

**returns** $results$
```