Zach Domke
5/6/18
CIS 315
Chris Wilson

```
1. 0| 0| ∞| ∞| ∞|−1| ∞|          1| 0| ∞| ∞| ∞|−1| ∞|
    | 1| 0| ∞| 2| ∞| ∞|           | 1| 0| ∞| 2| 0| ∞|
    | ∞| 2| 0| ∞| ∞|−8|           | ∞| 2| 0| ∞| ∞|−8|
    |−4| ∞| ∞| 0| 3| ∞|           |−4| ∞| ∞| 0|−5| ∞|
    | ∞| 7| ∞| ∞| 0| ∞|           | ∞| 7| ∞| ∞| 0| ∞|
    | ∞| 5|10| ∞| ∞| 0|           | ∞| 5|10| ∞| ∞| 0|

   2| 0| ∞| ∞| ∞|−1| ∞|          3| 0| ∞| ∞| ∞|−1| ∞|
    | 1| 0| ∞| 2| 0| ∞|           | 1| 0| ∞| 2| 0| ∞|
    | 3| 2| 0| 4| 2|−8|           | 3| 2| 0| 4| 2|−8|
    |−4| ∞| ∞| 0|−5| ∞|           |−4| ∞| ∞| 0|−5| ∞|
    | 8| 7| ∞| 9| 0| ∞|           | 8| 7| ∞| 9| 0| ∞|
    | 6| 5|10| 7| 5| 0|           | 6| 5|10| 7| 5| 0|

   4| 0| ∞| ∞| ∞|−1| ∞|          5| 0| 6| ∞| 8|−1| ∞|
    |−2| 0| ∞| 2|−3| ∞|           |−2| 0| ∞| 2|−3| ∞|
    | 3| 2| 0| 4| 2|−8|           | 3| 2| 0| 4| 2|−8|
    |−4| ∞| ∞| 0|−5| ∞|           |−4| 2| ∞| 0|−5| ∞|
    | 8| 7| ∞| 9| 0| ∞|           | 8| 7| ∞| 9| 0| ∞|
    | 6| 5|10| 7| 5| 0|           | 6| 5|10| 7| 5| 0|

   6| 0| 6| ∞| 8|−1| ∞|
    |−2| 0| ∞| 2|−3| ∞|
    |−2|−3| 0|−1|−3|−8|
    |−4| 2| ∞| 0|−5| ∞|
    | 8| 7| ∞| 9| 0| ∞|
    | 6| 5|10| 7| 5| 0|
```

```
2. oneStop(){
        array[V]
        grid[V][V]
        for u in V
             for v in V
                  for m in V
                       if pathExist(u,m) && pathEkists(m,v)
                            utom = zeroCap(u,m)
                            mtov = zeroCap(m,v)
                            array[m] = utom + mtov
```

```
                    else
                         array[m] = 0
               grid[u][v] = max{array[]}
     return grid
}
pathExists(u, v){
     if G[u][v] != ∞
          return TRUE
     return FALSE
}
zeroCap(u, v){
     return max{W*[u][v] | u,v in V}
}
```

3.

a. At our current position, we will determine the distance to travel from our current spot to all other spots in the forward direction. We then pick the position closest to 200 miles away.
b. We will start at position a_0. We will determine position a_a where $|a_a - a_0|$ is closest to 200 and adding $|a_a - a_0|$ to our total penalty. Now we move from a_a to a_b where $|a_b - a_a|$ is closest to 200 and adding $|a_b - a_a|$ to our total penalty. We continue this until we reach a_n.

4.

a. We will determine whether it is more beneficial to stay at our current position or take the penalty of moving plus the cost to stay at the other position.
b. Starting at the city with the lowest cost. The next position will be the minimum of either a) the cost of the current city or b) the cost of the other city plus the cost to change cities. From position 2 we will take the same minimum until we reach the last position. We will also run this algorithm starting at the other city, just in case it is smaller. We then take the minimum of the 2 plans.