Homework #3

1. a.) First, we must consider a cost array which is initialized to 0. Then we loop through the stops from the second stop to n. In each iteration, we set the minimum to the cost from the first stop to i, the current iteration count. Then, in each iteration we loop from the second stop to the current iteration – 1 and check to see if the last stored cost plus the cost from stop j to i is less than our previously set min. If so, we set the min equal to the stored cost plus the cost from stop j to i. Next, we save the cost for the inner iteration to the min value. The final stored cost represents the cheapest cost from i to j.

PSEUDOCODE  
The following was modified from: https://people.ucsc.edu/~ptantalo/cmps201/Spring10/hw9.doc

CanoeCost(R)  
 n = number of rows in R  
 C[1] = 0 # Cost initialized to 0  
 for i = 2 to n  
 min = R[1, i] # Set minimum to i stop cost  
 for j = 2 to i - 1  
 if C[j] + R[j, i] < min # If the previous cost + the cost from j to i is less than min  
 min = C[j] + R[j, i] # Set min to the cost + cost from j to i  
 C[i] = min # Store the minimum for next iteration  
 return C[n]

The recursive formula for filling in the array is as follows: