Assignment 4

Read sections 3.2, 3.3, 3.5, 3.6, 3.7.

Write two programs:

(1) solve the robot coin collection problem (from class) in the case where there are obstacles (squares the robot cannot pass through.) Here is an example input, where x denotes impassable squares:

```
C = [[0,x,0,1,0,0],

     [1,0,0,x,1,0],

     [0,1,0,x,1,0],

     [0,0,0,1,0,1],

     [x,x,x,0,1,0]]
```

As a starting point, here is the robot coin collection code from class:

```
def robot_coin_collection(C):
    rows = len(C)
    cols = len(C[0])
    F = [[C[0][0]]]
    for j in range(1, cols):
        F[0].append(F[0][j-1] + C[0][j]) # fill in first row
    for i in range(1, rows):
        F.append([F[i-1][0] + C[i][0]]) # fill in first col
        for j in range(1, cols):
            F[i].append(max(F[i-1][j], F[i][j-1]) + C[i][j]) # apply
recursion
    for k in F:
        print(k)
    return F[rows-1][cols-1]

C = [[0,0,0,0,1,0],
     [0,1,0,1,0,0],
     [0,0,0,1,0,1],
     [0,0,1,0,0,1],
     [1,0,0,0,1,0]]

print(robot_coin_collection(C))
```

(2) Implement algorithm 3.11 and test it on a few problem instances. As you implement it, you may modify the algorithm to improve it if you want. I'll introduce and motivate the Traveling Salesman Problem and discuss the algorithm in class on Tuesday.