CSC 440 - Assignment 8

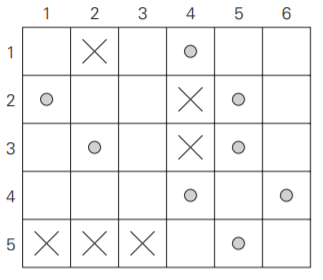
# Instructions

First, make a copy of this document (click on File → Make a copy). You will be editing your own copy of the assignment.

When you finish the assignment, you must download your copy and submit the downloaded file into Moodle (click on File → Download → File type; please use .doc, .docx, .pdf, or .rtf formats for your submissions.)

# Problems

1. (15 pts) How would you modify the dynamic programming algorithm for the **Coin Collecting Problem** if some cells on the board are inaccessible for the robot? Apply your algorithm to the board below, where the inaccessible cells are shown by X’s. Draw an optimal path on the board below once you have found one.



1. (15 pts) Solve the **Coin Change Problem** for n = 10 with denominations 1, 2, 4, and 8. Remember, this problem asks you to find the *smallest number of coins* you can use to make each total amount.

| n | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| F | 0 |  |  |  |  |  |  |  |  |  |  |

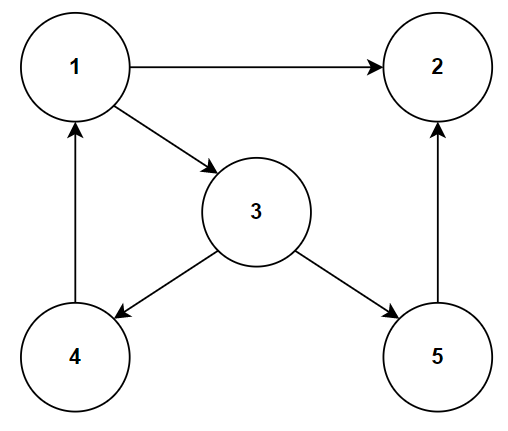
1. (15 pts) Solve the following instance of the **Knapsack Problem** using dynamic programming, with maximum capacity = 6.

| w1 = 2 | v1 = 4 |
| --- | --- |
| w2 = 3 | v2 = 7 |
| w3 = 1 | v3 = 3 |
| w4 = 2 | v4 = 6 |
| w5 = 4 | v5 = 16 |

Capacity *j*

| *i* | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 |  |  |  |  |  |  |
| 2 | 0 |  |  |  |  |  |  |
| 3 | 0 |  |  |  |  |  |  |
| 4 | 0 |  |  |  |  |  |  |
| 5 | 0 |  |  |  |  |  |  |

1. (15 pts) Apply **Warshall’s Algorithm** to compute the transitive closure of the following graph. Since creating and displaying several matrices may be tedious, you may instead list ordered pairs ( (i, j) coordinates in the matrix ) in the order in which they are changed from 0 to 1 according to the progression of the algorithm.



1. (**Extra Credit** 5 pts) There is a measure of difference between two strings commonly called “edit distance.” A popular approach to calculating this metric involves dynamic programming. Research this approach for calculating the edit distance between two strings, and write a paragraph describing how the algorithm works, and specifically how it leverages dynamic programming to accomplish its goal.