1. Solve the instance 5, 1, 2, 10, 6 of the coin-row problem.

2. How would you modify the dynamic programming algorithm for the coincollecting 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. How many optimal paths are there for this board?
3 Design a dynamic programming algorithm for the following problem. Find the maximum total sale price that can be obtained by cutting a rod of \( n \) units long into integer-length pieces if the sale price of a piece \( i \) units long is \( P_i \) for \( i=1,2,3,\ldots,n \) What are the time and space efficiencies of your algorithm?

4. Write pseudocode of the greedy algorithm for the change-making problem, with an amount \( n \) and coin denominations \( d_1 > d_2 > \cdots > d_m \) as its input. What is the time efficiency class of your algorithm?

5. Consider the problem of scheduling \( n \) jobs of known durations \( t_1, t_2, \ldots, t_n \) for execution by a single processor. The jobs can be executed in any order, one job at a time. You want to find a schedule that minimizes the total time spent by all the jobs in the system. (The time spent by one job in the system is the sum of the time spent by this job in waiting plus the time spent on its execution.)
Design a greedy algorithm for this problem. Does the greedy algorithm always yield an optimal solution?