Question 1

There is an $n \times n$ grid of squares. Each square is either *special*, or has a positive integer *cost* assigned to it. No square on the border of the grid is special.

A set of squares S is said to be *good* if it does not contain any special squares and, starting from any special square, you cannot reach a square on the border of the grid by performing up, down, left and right moves without entering a cell belonging to S.

1.1 [**5 marks**] In the following diagram, squares are labelled 'X' if they are special, otherwise they are labelled by their cost. Identify a good set of squares with minimum total cost for this particular grid.

5	3	4	9
4	X	3	6
1	9	X	4
1	2	3	5

Your answer here.

1.2 [**15 marks**] Design an algorithm which receives an arbitrary $n \times n$ grid, runs in time polynomial in n and determines a good set of squares with minimum total cost.

Your answer here.