Evan Zhang | z5383657

Question 1

1.1

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

Total cost = 20

Good set = {3, 3, 4, 3, 2, 1, 4}

1.2

Begin by constructing a flow network as follows:

- Vertices:
  - Source s and sink t
  - For each square, say with coordinates (i,j), construct a pair of vertices $v_{i,j}^{in}$ and $v_{i,j}^{out}$ and place an edge of capacity equal to the square (i,j)'s weight, representing a vertex capacity equal to a square's weight for a square (i,j).
    - Do not construct vertices for corner squares.
    - For each special square, construct a single vertex $v_{i,j}$.
- Edges:
  - Connect s to each special square vertex with capacity infinity.
  - Connect each special square to in-vertices of cardinally adjacent squares.
    - Do not construct edges for adjacent special squares.
  - For each pair of adjacent squares (i, j) and (i', j'), place edges of capacity infinity from $v_{i,j}^{out}$ to $v_{i',j'}^{in}$ and from $v_{i',j'}^{out}$ and $v_{i,j}^{in}$.
    - Do not construct edges between border square vertices.
  - Connect all border square vertices to t with capacity infinity.

From this flow network construction, we run Edmonds-Karp to find the maximum flow and the minimum cut, separating all vertices into either set S or set T, depending on whether they are closer to the source or the sink respectively. Add all squares which have an edge crossing the minimum cut to the set of 'good' squares.

We have constructed the graph so all 'paths' to the border are 'bottlenecked' by the minimum value across that path. When max flow is achieved, these bottlenecks flow at their max capacity and form the vertex border from S to T (i.e., have an edge crossing the minimum cut) and we can simply add all vertices with an edge starting in S and ending in T to the set of 'good' squares.

The time complexity is $O(VE^2)$ where k = num_special_squares ≤ n, $V \leq 2n^2 + 2$ and $E \leq 5k + n^2 + 4(n-2)^2 + 4(n-2)$, so the algorithm runs in time polynomial in n as required.