Question 2 Solution

a) We will first decide on the notation that we will use for the described quatitites. Nodes will be repsesented as (i, j) pairs, for $0 \leq i, j \leq n$, and the weight on edge $(v \rightarrow u)$ will be denoted by w_v^u. Moreover, we will let $OPT(i, j)$ represent the maximum weight of paths from $(0, 0)$ to (i, j).

First, as base case, we will note that $OPT(0, 0) = 0$ and $OPT(0, j) = \sum_{k=1}^{j} w_{(0,k-1)}^{(0,k)}$, and $OPT(i, 0) = \sum_{k=1}^{i} w_{(i-1,0)}^{(i,0)}$, since there is always a unique path to these nodes.

Next point is that the only way that we can get from $(0, 0)$ to (i, j), for $1 \leq i, j \leq n$ is using one of the two edges : $(i-1, j) \rightarrow (i, j)$ or $(i, j-1) \rightarrow (i, j)$.

Also, if we have a maximum path to (i, j) that goes through $(i, j-1) \rightarrow (i, j)$, then the rest of the path has to be optimal too. If that would not be the case, then we could simply improve the optimal path to (i, j) by improving the subpath to $(i, j-1)$. The same hold for $(i-1, j) \rightarrow (i, j)$.

With this being said, it becomes clear that $OPT(i, j)$ will be:

- $\max\left\{OPT(i-1, j) + w_{(i-1,j)}^{(i,j)}, OPT(i, j-1) + w_{(i,j-1)}^{(i,j)}\right\}$ for $i, j \geq 1$
- $\sum_{k=1}^{j} w_{(0,k-1)}^{(0,k)}$ for $j \geq 1$
- $\sum_{k=1}^{i} w_{(k-1,0)}^{(k,0)}$ for $i \geq 1$
- 0 for $i = j = 0$

Now, it is not enough just to state what the recurrence relation is. First, it is important to choose the right order of computation in an efficient way. There are two alternatives. One, is to follow the recurrence and to realize that if we complete the matrix row by row, starting at $(0, 0)$, then we are in business. Second, is to start at (n,n) and recursivelly to complete only the necessary entries in the matrix. The recurrsion guarantees that our recursion will not unfold more than $2n$ times, and since we memoize our values, we avoid having an exponetial recursion.

We are not finished yet. Although we have completed the maxtrix, we don't have what we are looking for: the optimal path. One way is to remember at each step which of the choices maximized the weight, and then starting at (n, n) to use these choices as a way to go backwards to $(0, 0)$. Still, it is not necessary to memorize these choices. Since we have the entrire matrix, we could compare $OPT(i, j)$ to $OPT(i, j-1)$ and $OPT(i-1, j)$, and deduce which one maximized the recurrence relation.

b) The most popular answer is to negate all weights in the graph. Still, you should be careful since this does not work in general. So, "obviously the max weight path is the min weight path for the original problem" is not enough as a proof. What makes it work in the given problem is the fact that all paths have the same length. When the length of paths is bounded, then we have no issue with infinite weight paths, so negating all weights in the graph, simply negates the weight of each path.

Although the above is safe anytime there are no cycles, there were other correct proposed solutions. We could also substract each weight from the maximum weight of the edges. Note that this involves scaling, which only works, as is the case here, when the length of all paths is the same($2n$ here).

Other popular incorrect solution was to claim that if the maximal path chooses a vertical edge, the minimal should choose the right. Notice that this is wrong since it easily be the case that both the maximal and the minimal have edges in common. To convince yourself, I provide bellow such an example:

$$\begin{array}{ccccc}
\bullet & \rightarrow 0 \rightarrow & \bullet & \rightarrow 1 \rightarrow & \bullet \\
\downarrow & & \downarrow & & \downarrow \\
2 & & 0 & & 1 \\
\downarrow & & \downarrow & & \downarrow \\
\bullet & \rightarrow 1 \rightarrow & \bullet & \rightarrow 1 \rightarrow & \bullet \\
\downarrow & & \downarrow & & \downarrow \\
2 & & 0 & & 1 \\
\downarrow & & \downarrow & & \downarrow \\
\bullet & \rightarrow 2 \rightarrow & \bullet & \rightarrow 0 \rightarrow & \bullet
\end{array}$$