AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

22BIO211: Intelligence of Biological Systems - 2

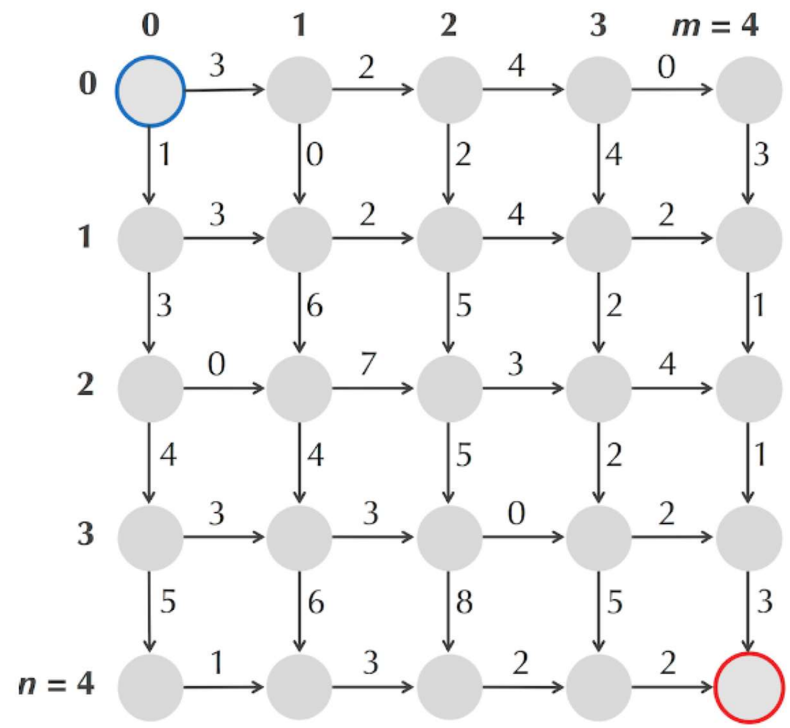
MANHATTAN TOURIST PROBLEM REVISITED

Dr. Manjusha Nair M
Amrita School of Computing, Amritapuri

Email : manjushanair@am.amrita.edu
Contact No: 9447745519

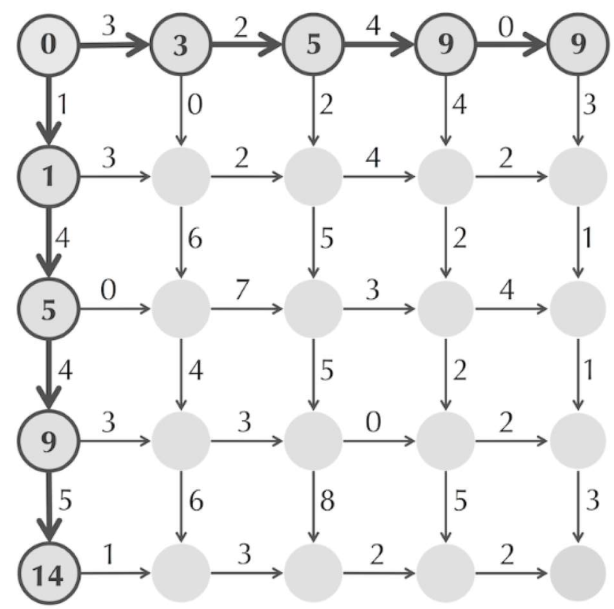
The Manhattan Tourist Problem Revisited

- What is the length of the longest path between the source (0,0) and sink (n,m)?
- Use dynamic programming i.e, solve each of the smaller problems once rather than billions of times
- To find the length of the longest path from source (0, 0) to sink (n, m)
  - *we will first find the lengths of the longest paths from the source to all nodes (i, j) in the grid, expanding slowly outward from the source.*



The Manhattan Tourist Problem Revisited

- We will henceforth denote the length of the longest path from (0, 0) to (i, j) as $s_{i,j}$.
- Computing $s_{0,j}$ (for $0 \le j \le m$) is easy, since we can only reach (0, j) by moving right (→) and do not have any flexibility in our choice of path.
  - *Thus, $s_{0,j}$ is the sum of the weights of the first j horizontal edges leading out from the source.*
- Similarly, $s_{i,0}$ is the sum of the weights of the first i vertical edges from the source.



The Manhattan Tourist Problem Revisited

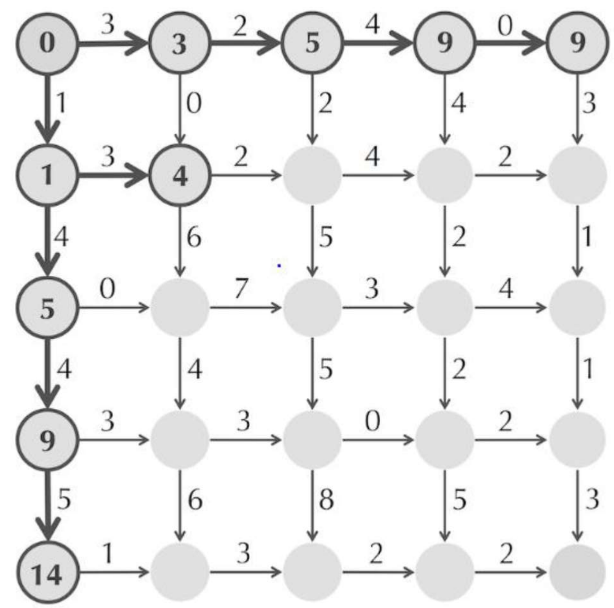
- For $i > 0$ and $j > 0$, the only way to reach node (i, j) is by moving down from node $(i - 1, j)$ or by moving right from node $(i, j - 1)$.
- Thus, $s_{i,j}$ can be computed as the maximum of two values:
  - $s_{i-1,j}$ + *weight of the vertical edge from (i - 1, j) to (i, j)*
  - $s_{i,j-1}$ + *weight of the horizontal edge from (i, j - 1) to (i, j)*
- Now that we have computed $s_{0,1}$ and $s_{1,0}$, we can compute $s_{1,1}$.
- You can arrive at (1, 1) by traveling down from (0, 1) or right from (1, 0).

The Manhattan Tourist Problem Revisited

- Therefore, $s_{1,1}$ is the maximum of two values:
  - $s_{0,1}$ + *weight of the vertical edge from (0, 1) to (1, 1)*

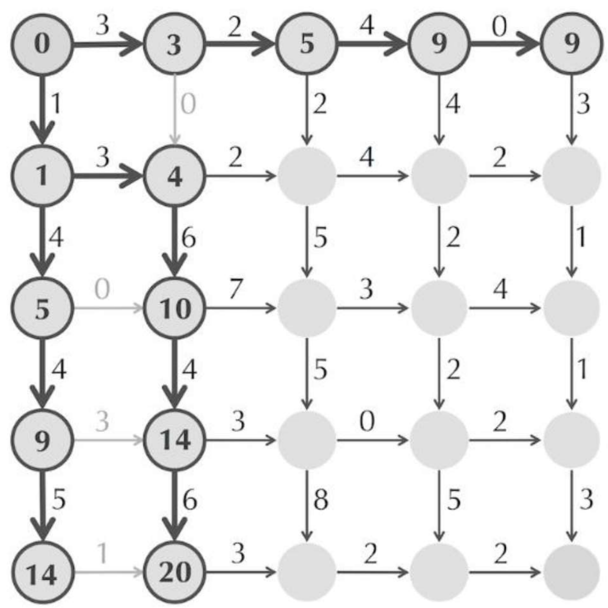
    $= 3 + 0 = 3$
  - $s_{1,0}$ + *weight of the horizontal edge from (1, 0) to (1, 1)*

    $= 1 + 3 = 4$
- Since our goal is to find the longest path from (0, 0) to (1, 1), we conclude that $s_{1,1} = 4$.
- Because we chose the horizontal edge from (1, 0) to (1, 1), the longest path through (1, 1) must use this edge, which we highlight in the figure.



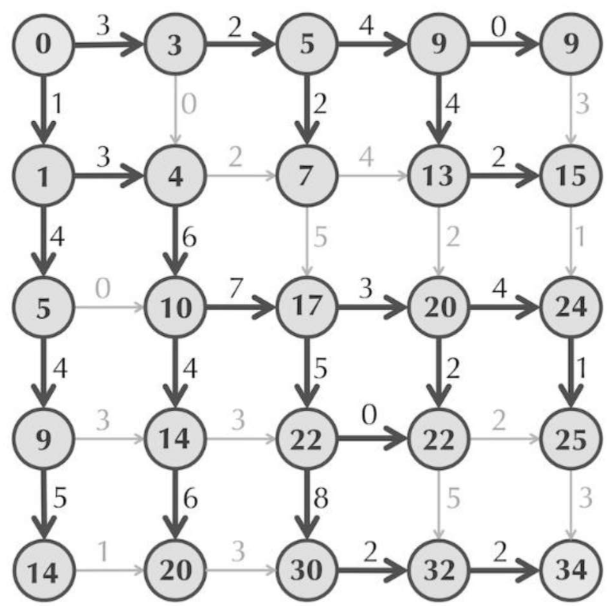
The Manhattan Tourist Problem Revisited

- Similar logic allows us to compute the rest of the values in column 1;
  - *for each $s_{i,1}$, we highlight the edge that we chose leading into (i, 1), as shown in the figure below.*



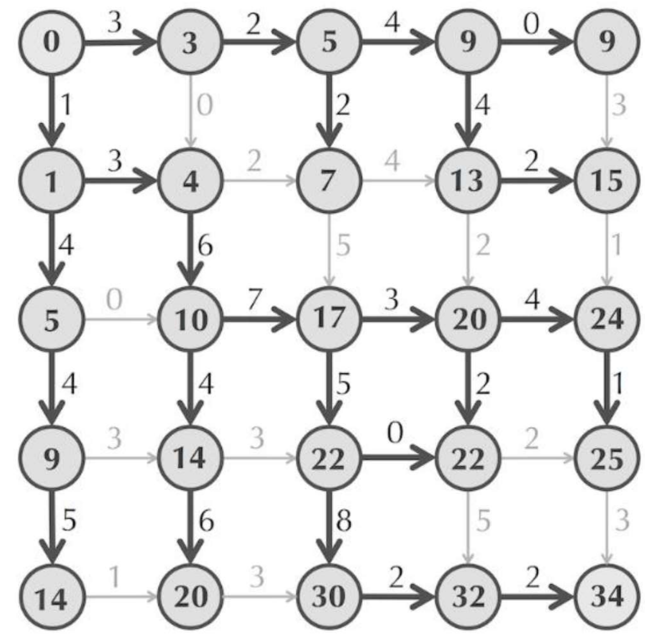
The Manhattan Tourist Problem Revisited

- Continuing column-by-column, we can compute every score $s_{i,j}$ in a single sweep of the graph, eventually calculating $s_{4,4} = 34$.



The Manhattan Tourist Problem Revisited

- For each node (i, j), we will highlight the edge leading into (i, j) that we used to compute $s_{i,j}$. However, note that we have a tie when we compute $s_{3,3}$.
- To reach (3, 3), we could have used either the horizontal or vertical incoming edge, and so we will highlight both of these edges in the completed graph



$$s_{3,3} = \max \begin{cases} s_{2,3} + \text{ weight of vertical edge from } (2, 3) \text{ to } (3,3) = 20 + 2 = 22 \\ s_{3,2} + \text{ weight of horizontal edge from } (3, 2) \text{ to } (3,3) = 22 + 0 = 22 \end{cases}$$

Find Longest Path

- How could you use the highlighted edges in the figure from the previous step, to reconstruct a longest path?
- Code Challenge: Find the length of a longest path in the Manhattan Tourist Problem.
  - *Input: Integers n and m, followed by an $n \times (m + 1)$ matrix Down and an $(n + 1) \times m$ matrix Right. The two matrices are separated by the "-" symbol.*
  - *Output: The length of a longest path from source $(0, 0)$ to sink (n, m) in the rectangular grid whose edges are defined by the matrices Down and Right.*

Find Longest Path – Dynamic Programming Algorithm

ManhattanTourist(n, m, $Down$, $Right$)

$s_{0,\,0} \leftarrow 0$

for $i \leftarrow 1$ **to** n

$s_{i,\,0} \leftarrow s_{i-1,\,0} + down_{i-1,\,0}$

for $j \leftarrow 1$ **to** m

$s_{0,\,j} \leftarrow s_{0,\,j-1} + right_{0,\,j-1}$

for $i \leftarrow 1$ **to** n

for $j \leftarrow 1$ **to** m

$s_{i,\,j} \leftarrow \max\{s_{i-1,\,j} + down_{i-1,\,j},\ s_{i,\,j-1} + right_{i,\,j-1}\}$

return $s_{n,\,m}$

Summary

- The Manhattan Tourist Problem Revisited
- Find Longest Path
  - *Dynamic Programming*