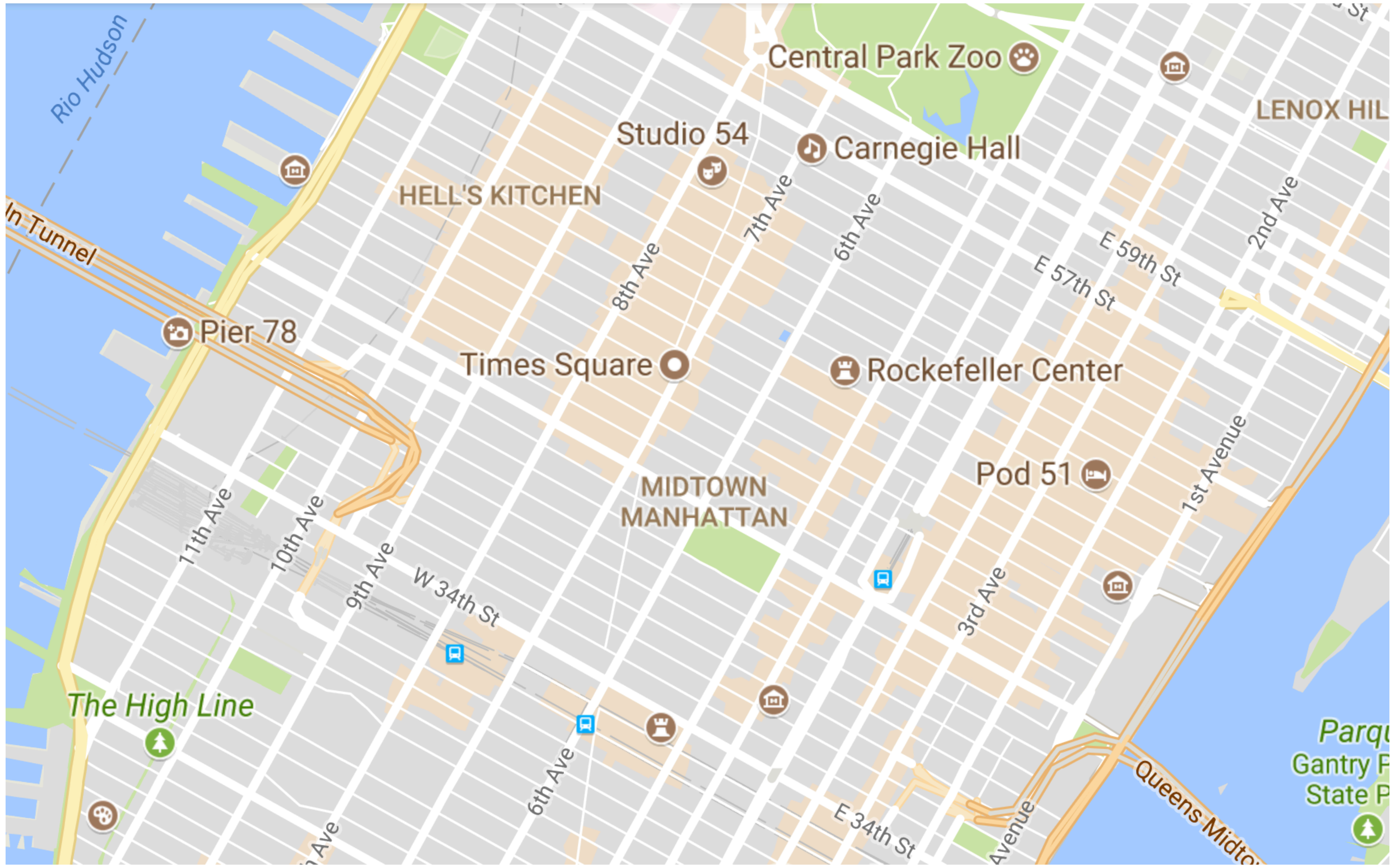
	0	1	2	3	4	5	6	7	8	9	10
0	*	←	*	←	*	←	*	←	*	←	*
1	↑	↖	↑	↖	↑	↖	↑	↖	↑	↖	↑
2	*	←	*	←	*	←	*	←	*	←	*
3	↑	↖	↑	↖	↑	↖	↑	↖	↑	↖	↑
4	*	←	*	←	*	←	*	←	*	←	*
5	↑	↖	↑	↖	↑	↖	↑	↖	↑	↖	↑
6	*	←	*	←	*	←	*	←	*	←	*
7	↑	↖	↑	↖	↑	↖	↑	↖	↑	↖	↑
8	*	←	*	←	*	←	*	←	*	←	*
9	↑	↖	↑	↖	↑	↖	↑	↖	↑	↖	↑
10	*	←	*	←	*	←	*	←	*	←	*

Profa. Dra. Raquel C. de Melo-Minardi
Departamento de Ciência da Computação
Instituto de Ciências Exatas
Universidade Federal de Minas Gerais

MÓDULO 4
ALGORITMOS PARA BIOINFORMÁTICA
O problema do turista em Manhattan

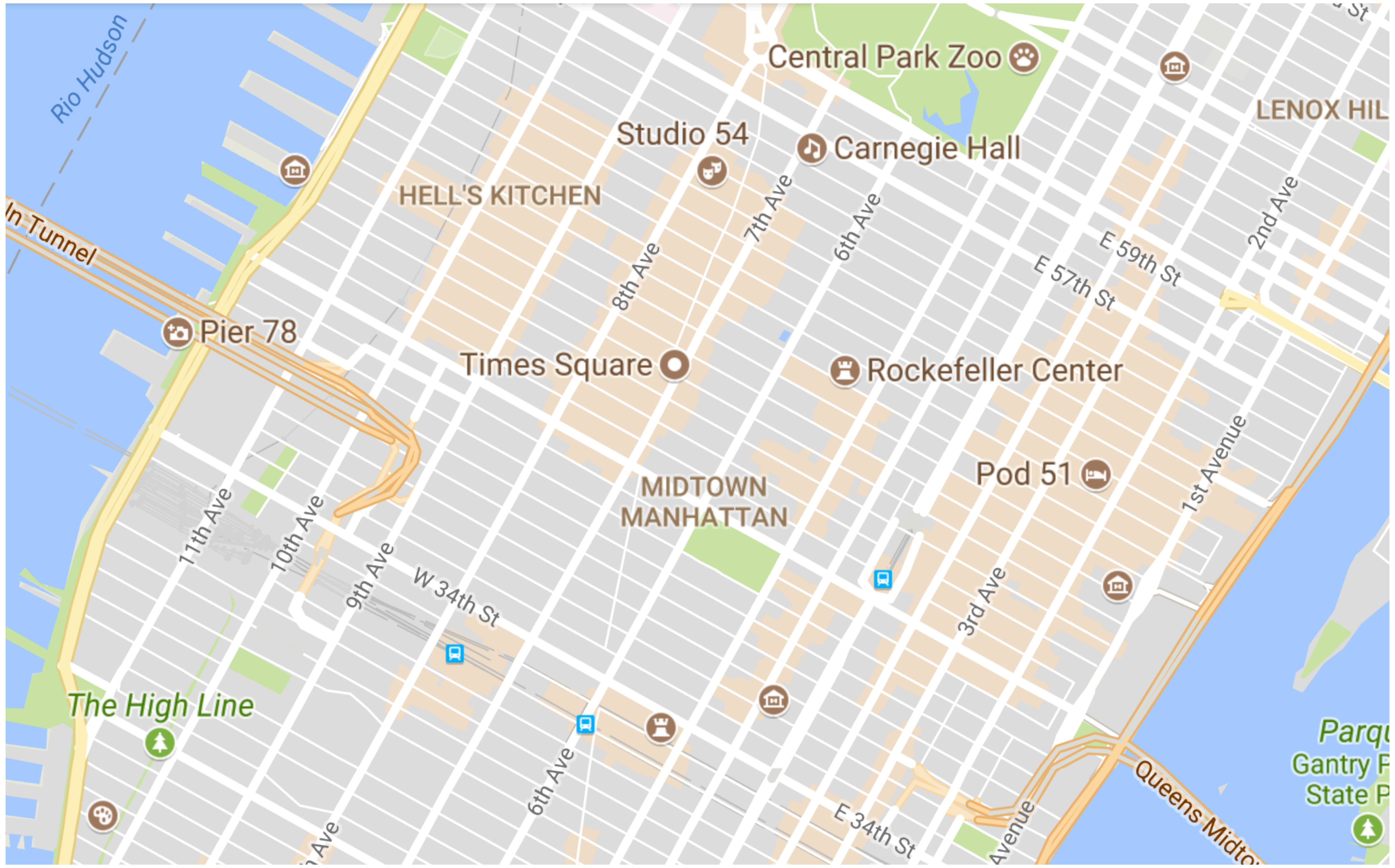
O PROBLEMA DO TURISTA EM MANHATTAN

- Antes de entrar no alinhamento de sequências, introduziremos outro problema que também é resolvido usando o mesmo paradigma: o problema do turista em Manhattan



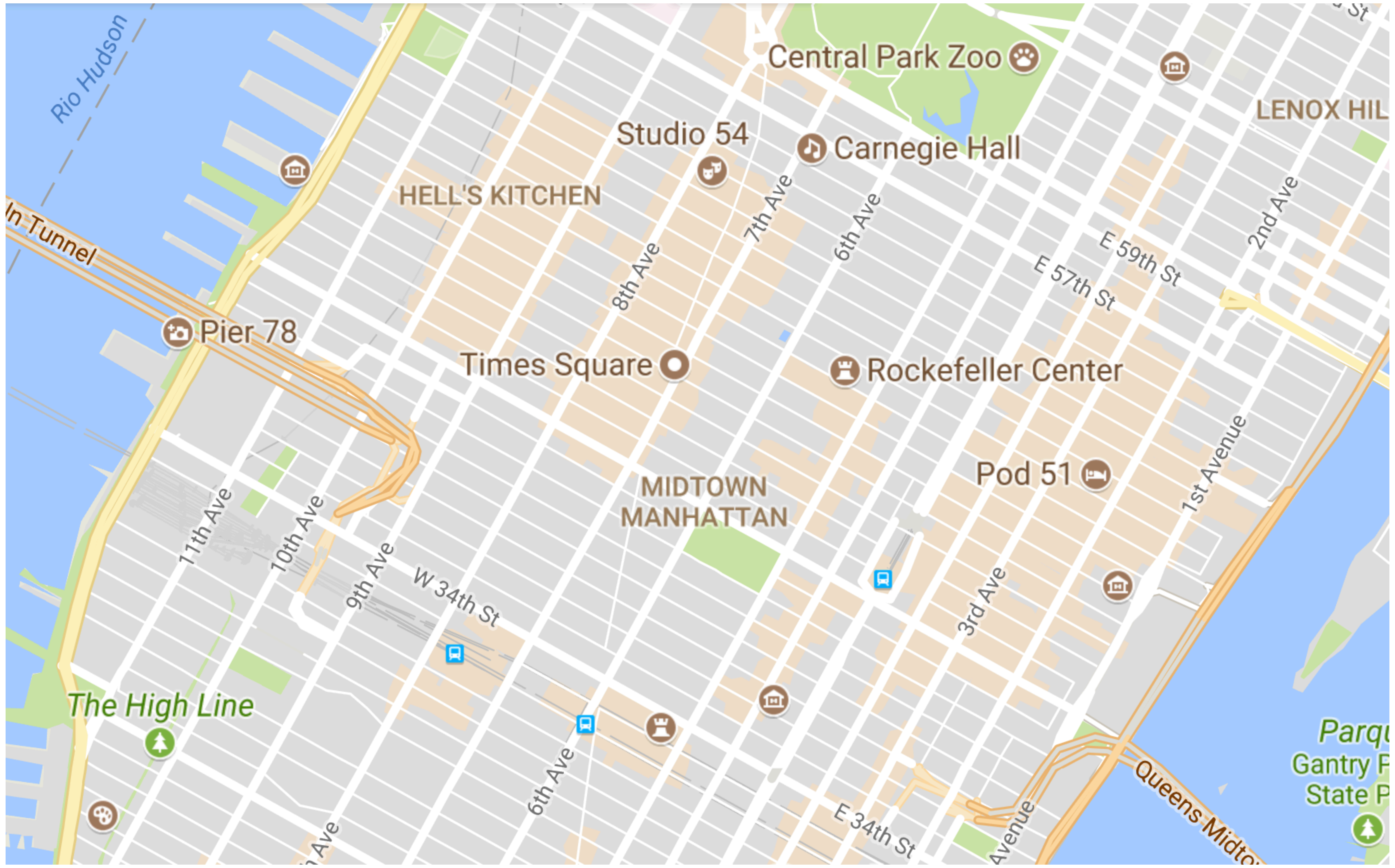
O PROBLEMA DO TURISTA EM MANHATTAN

- Esse é um mapa do centro de Manhattan onde se pode ver inúmeros pontos turísticos como o Carnegie Hall, o Rockfeller Center, a Times Square, entre outros. Uma característica interessante dessa região que se pode notar através do mapa é a organização bem retangular dos quarteirões



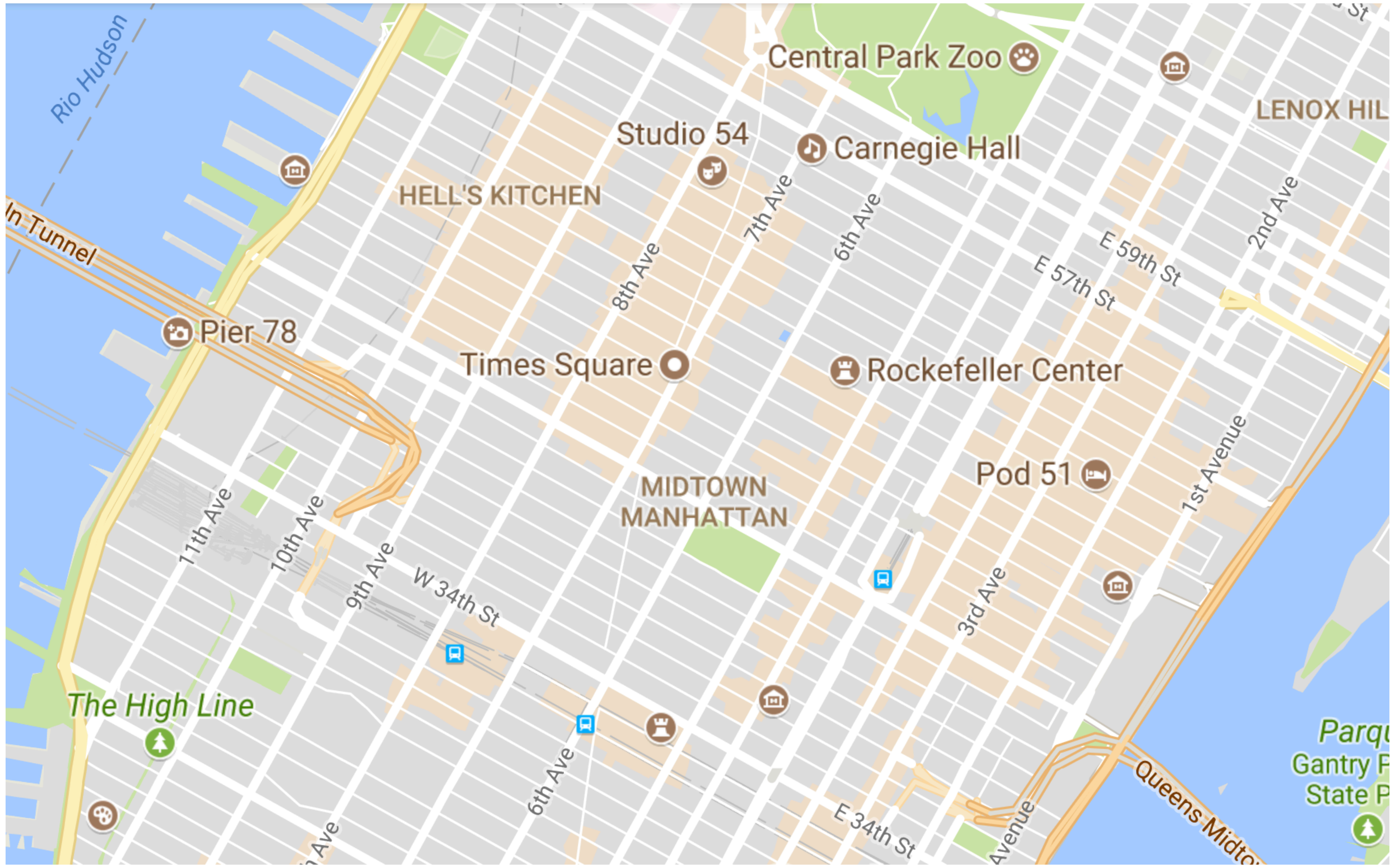
O PROBLEMA DO TURISTA EM MANHATTAN

- Você está em Manhattan e está em uma excursão que lhe deixará na esquina entre a 59th Street e a 8th Avenue e lhe buscará no Chrysler Building na esquina entre a 42nd Street e a Lexington Avenue



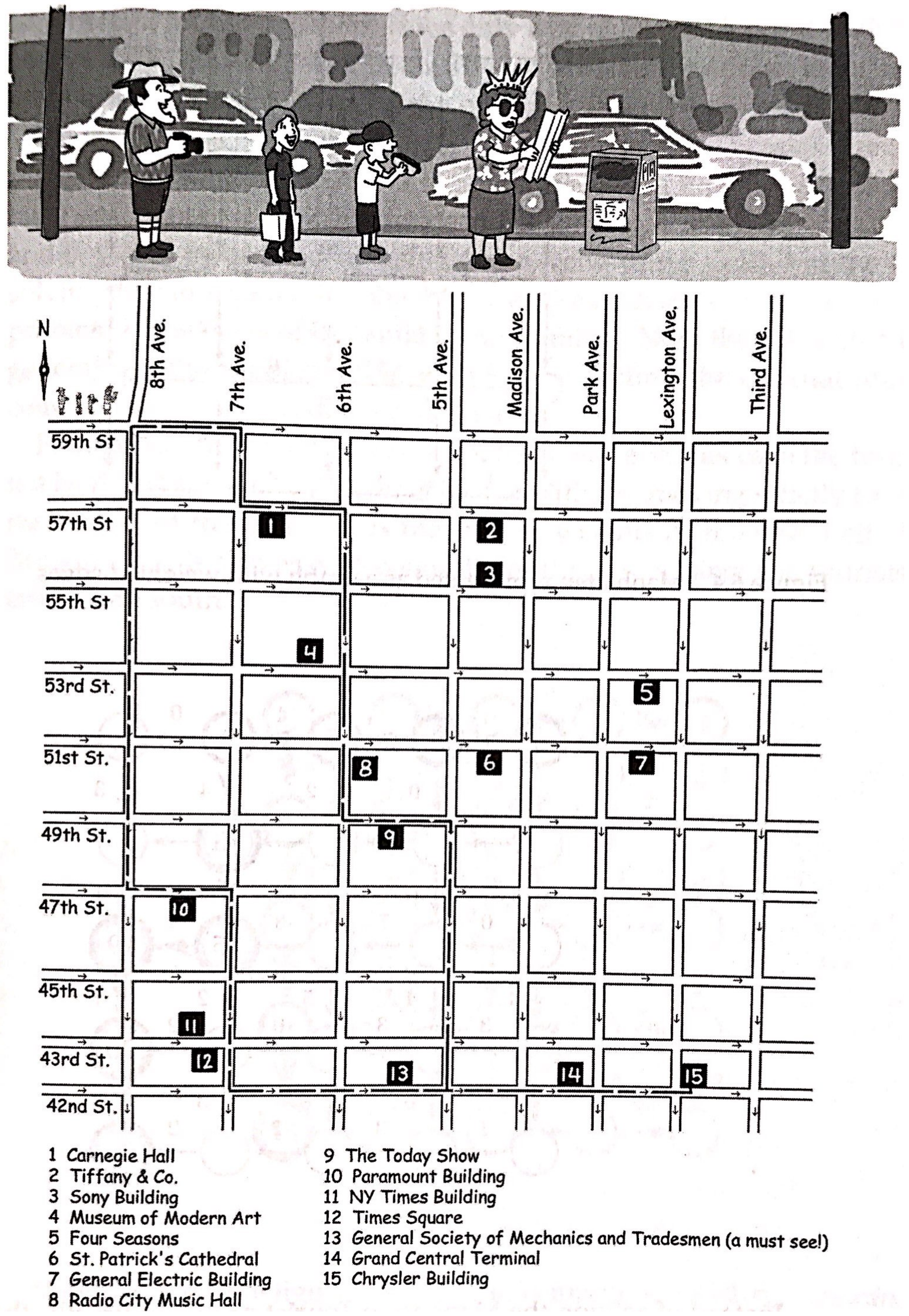
O PROBLEMA DO TURISTA EM MANHATTAN

- Há inúmeras atracões turísticas na região entre esses pontos e você deseja organizar seu trajeto de forma a visitar o máximo de pontos turísticos quanto possível sem se preocupar com o tempo



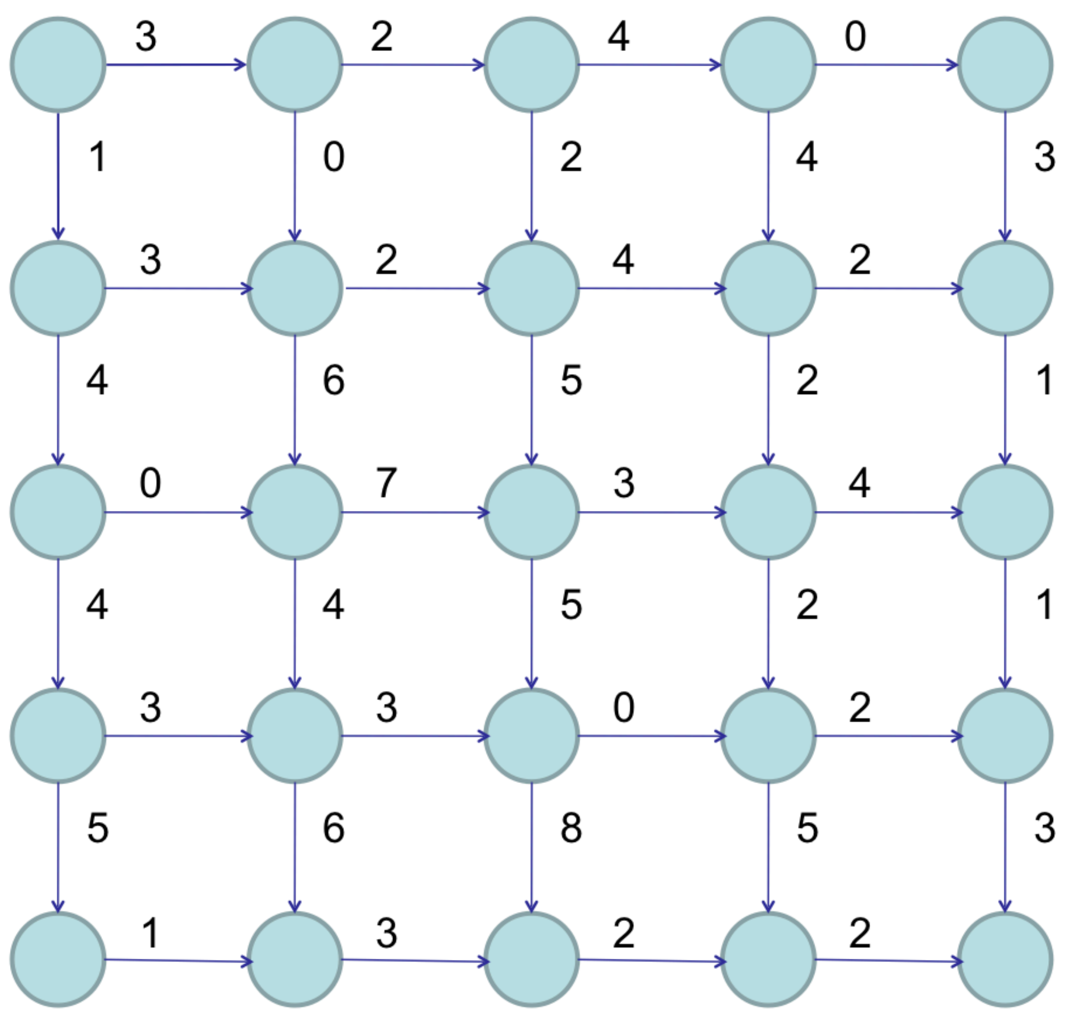
O PROBLEMA DO TURISTA EM MANHATTAN

- Veja um mapa mais simplificado retirado de [Jones e Pevzner, 2004] de onde todo esse exemplo foi retirado



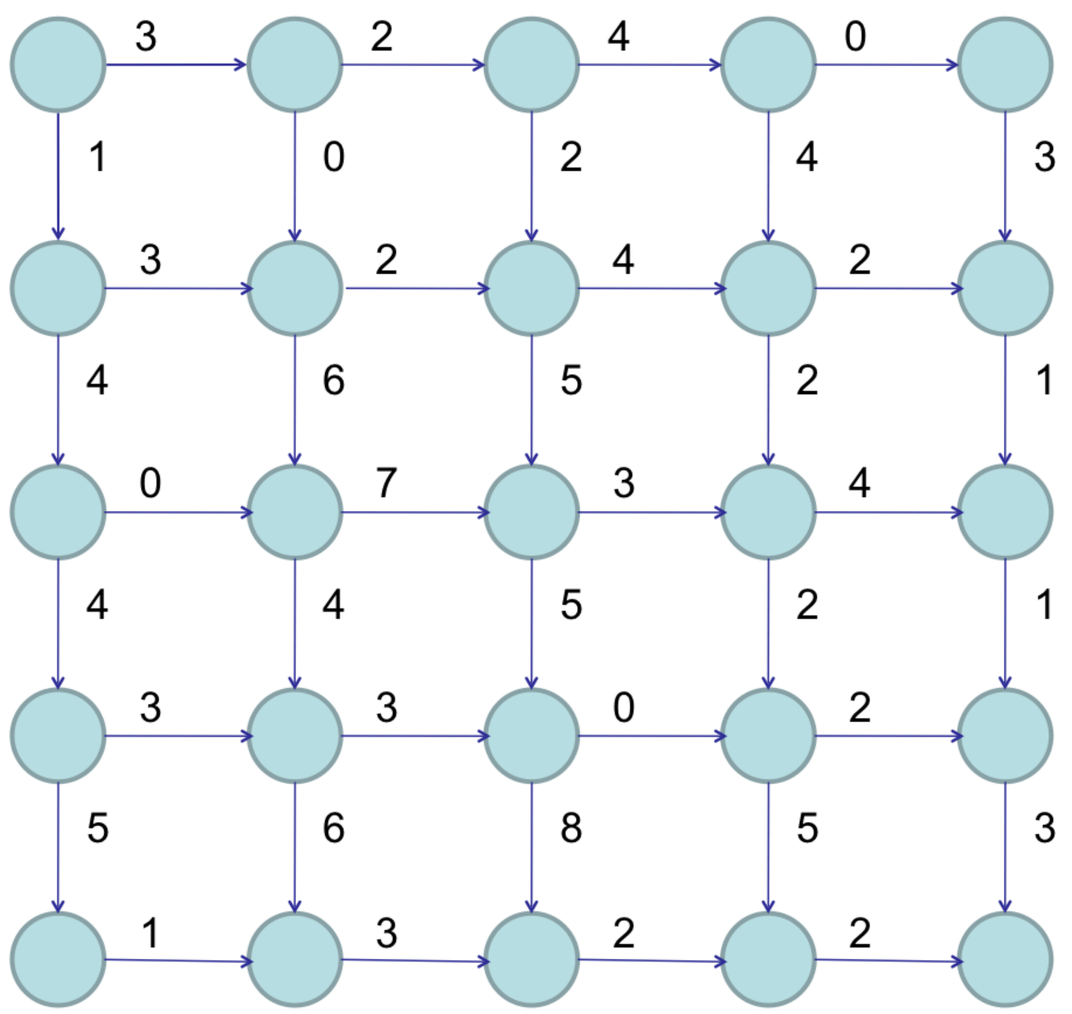
[Jones e Pevzner, 2004] Jones, Neil C., and Pavel Pevzner. An introduction to bioinformatics algorithms. MIT press, 2004.

MODELAGEM DO PROBLEMA



- Como todos os quarteirões são retangulares, essa região pode ser facilmente representada como um grid no qual
  - **nós** (círculos): as esquinas
  - **arestas** (setas): os trechos das ruas entre as esquinas
    - os rótulos das arestas indicam quantos pontos turísticas há naquele trecho da rua

MODELAGEM DO PROBLEMA



- O nó do canto superior esquerdo é a **fonte**
- O nó do canto inferior direito é o **sumidouro**
- O objetivo do turista é sair da fonte e chegar no sumidouro com o maior número possível de pontos turísticos visitados
- Ele só pode caminhar de cima para baixo e da esquerda para a direita não sendo permitidas voltas

Problema do turista em Manhattan

Encontre o caminho máximo em um grid rotulado

Entradas: Um grid rotulado G com pesos e dois vértices distintos: fonte e sumidouro
Saída: O caminho máximo em G entre fonte e sumidouro

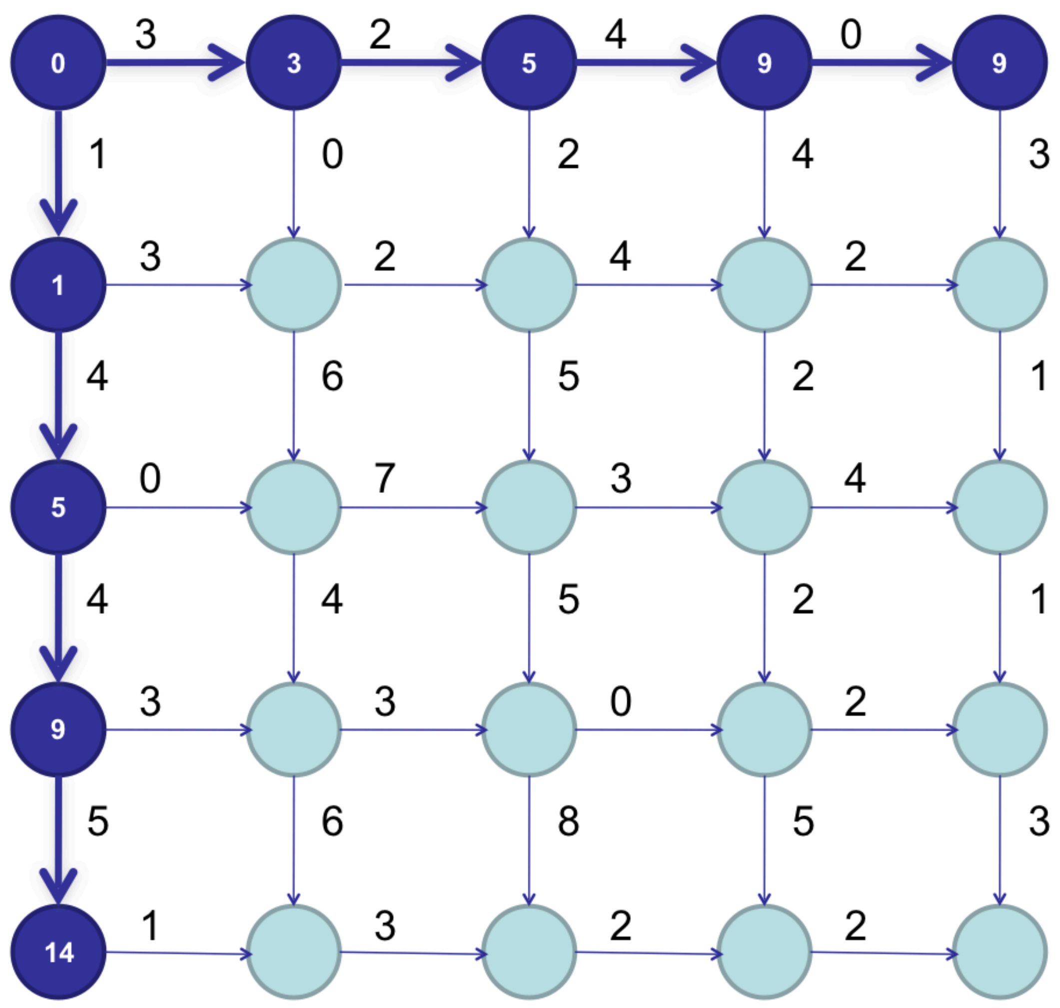
SOLUÇÃO

- Como você resolveria esse problema?
  - Usar um método de **força bruta** seria uma possibilidade e consistiria em buscar todos os possíveis caminhos entre fonte e sumidouro
    - Essa solução seria **proibitiva em termos de custo computacional** que seria fatorial
  - Poderíamos então usar uma abordagem “**gulosa**" que consiste em, a cada esquina, olhar as ruas adiante escolhendo a opção aparentemente mais vantajosa localmente dentre as duas possíveis ruas a tomar
    - Essa solução pode levar a uma solução ruim levando o turista para um local com poucas ou nenhuma atração

PROGRAMAÇÃO DINÂMICA

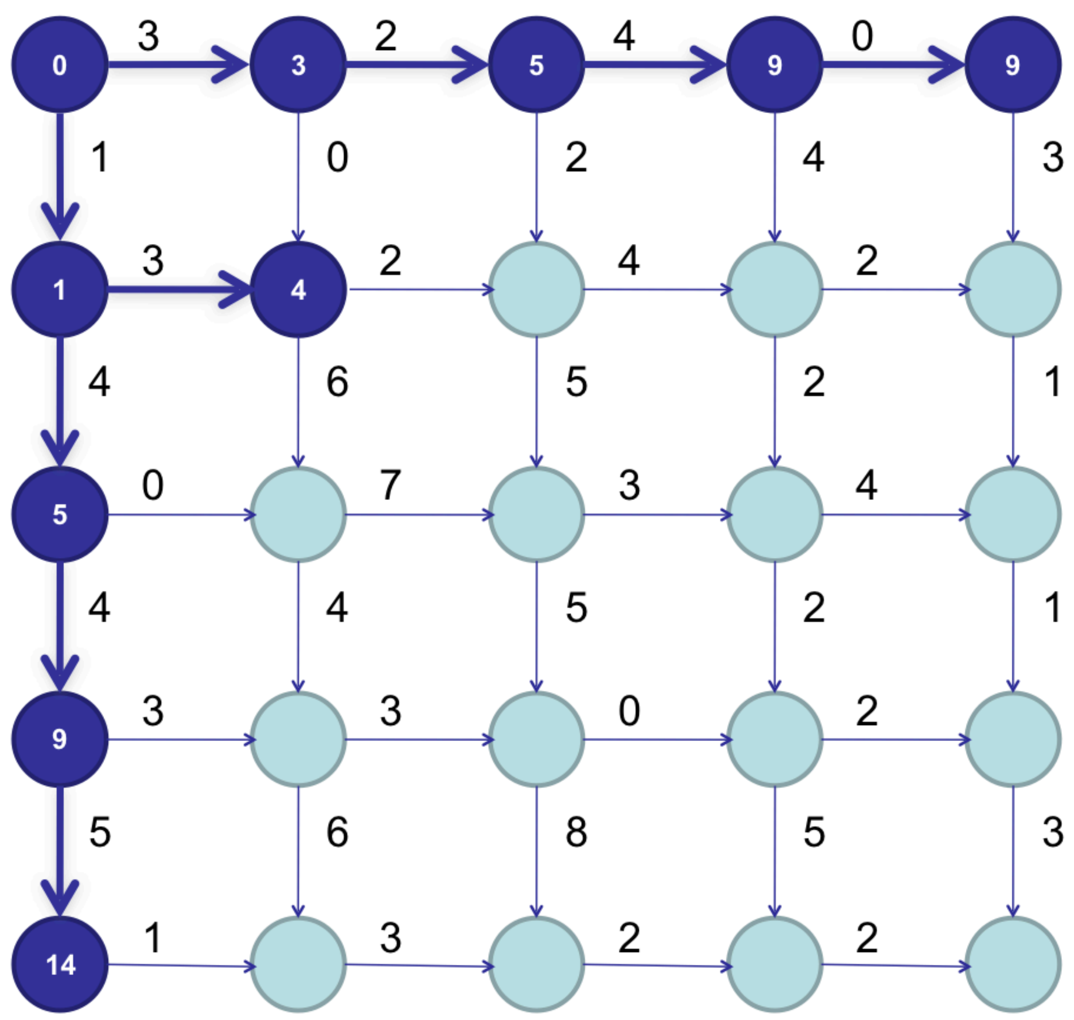
- Esse problema tem **subestrutura ótima** e pode ser resolvido por **programação dinâmica**
- Ao invés de solucionar o problema de ir da fonte *(0, 0)* ao sumidouro *(n, m)*, resolveremos o problema mais geral e encontrar o caminho máximo entre a fonte e um vértice arbitrário *(i, j)*, com $0 \leq i \leq n$, $0 \leq j \leq m$
- Denotaremos o menor caminho de $s_{i,j}$ até $s_{n,m}$ a solução do **Problema do Turista em Manhattan**
- Estamos escolhendo um problema geral ainda mais difícil mas isso não é verdade pois, para resolver o Problema do Turista em Manhattan, precisaremos resolver os subproblemas por programação dinâmica, então, é a mesma dificuldade

SOLUÇÃO



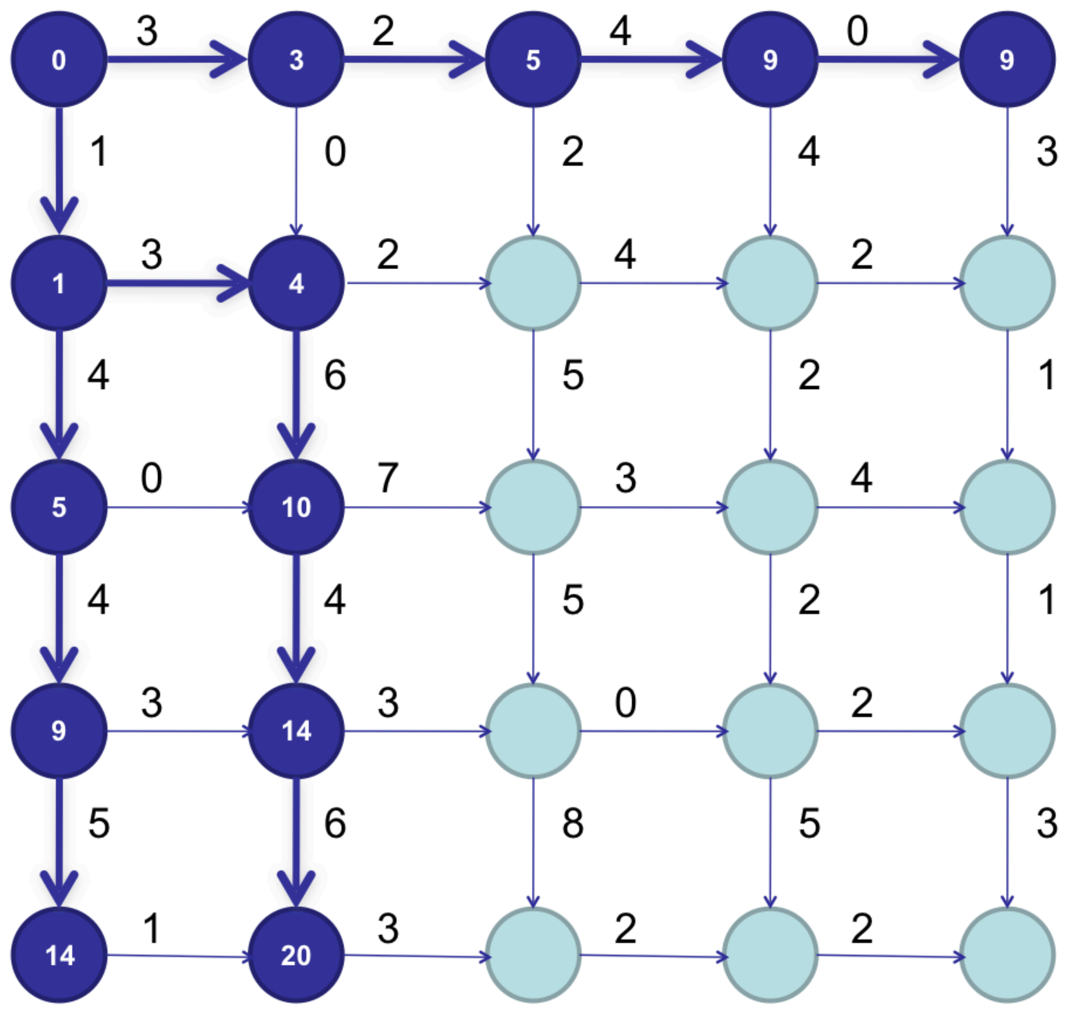
- Comecemos por $s_{0,j}$ (para $0 \leq j \leq m$) que é um problema simples bastando que o turista se mova sempre à direita e acumulando as pontuações
- A seguir, podemos fazer o mesmo para $s_{i,0}$ (para $0 \leq i \leq n$) em que o turista se moverá sempre para baixo e acumulando as pontuações
- Note que colocamos a soma da pontuação do caminho da fonte até o ponto $s_{i,j}$ como um rótulo dentro do nó

SOLUÇÃO



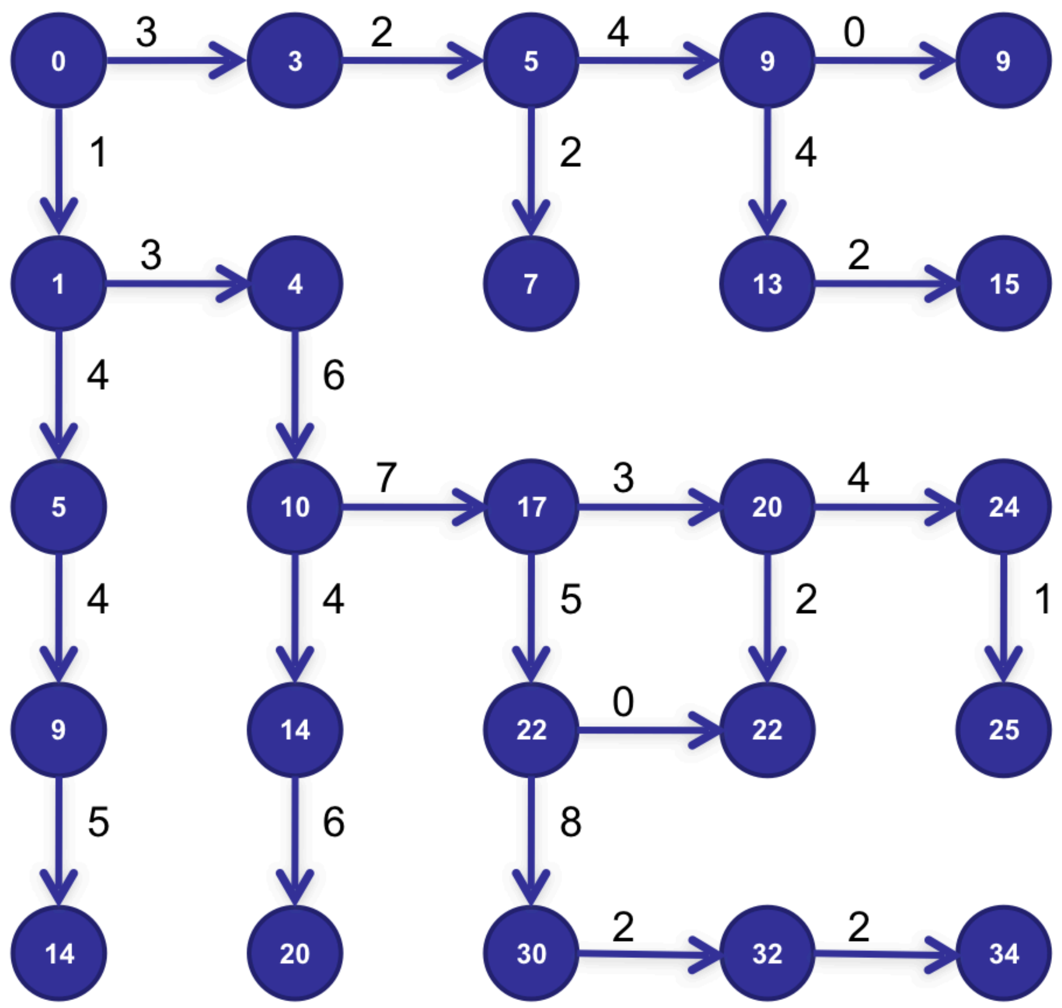
- A seguir, podemos preencher tanto a linha 2 quando a coluna 2 e assim sucessivamente
- Note que as novas soluções que são geradas para cada problema (cada nó é um tamanho de problema) são compostas pelas soluções dos subproblemas menores que já foram calculados
- Por exemplo, a célula $s_{1,1}$ usará as soluções $s_{0,1}$ ou $s_{1,0}$
  - A melhor solução será vir de $s_{1,0}$ somando o valor da aresta que é *3*

SOLUÇÃO



- Veja a extensão da solução para toda a segunda coluna

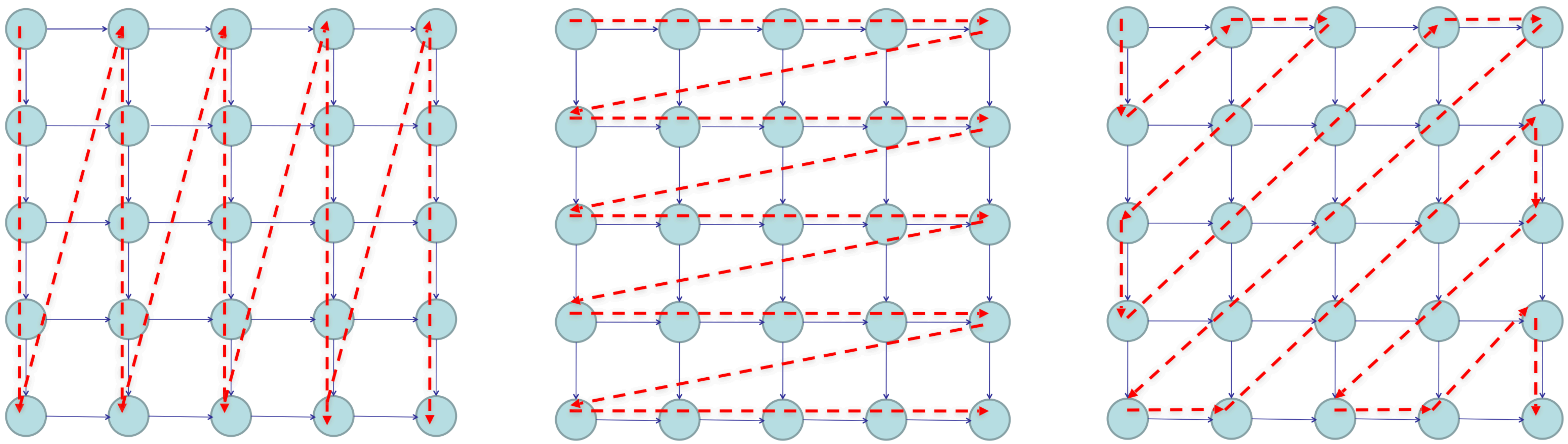
SOLUÇÃO



- Essa é a matriz completa com os caminhos máximos saindo da fonte $s_{0,0}$ para todos os possíveis $s_{i,j}$ com $0 \leq i \leq n$, $0 \leq j \leq m$

SOLUÇÃO

- Essa matriz pode ser preenchida em três possíveis sequências



- Não importa a ordem em que seja preenchida, seu conteúdo será o mesmo