What is wrong with the LCS scoring model?

```
YAFDLGYTCMFPVLLGGGELHIVQKETYTAPDEIAHYIKEHGITYIKLTPSLFHTIVNTASFAFDANFESIRLIVLGGEKIIPIDVIAFRKMYGHIE-FINHYGPTEATIGA
-AFDVSAGDFARALLTGGQLIVCPNEVKMDPASLYAIIKKYDITIFEATPALVIPLMEYI-YEQKLDISQLQILIVGSDSCSMEDFKTLVSRFGSTIRIVNSYGVTEACIDS
```

Marahiel's (biologically adequate) alignment

Longest Common Subsequence (higher #matches but biologically inadequate)

The optimal longest common subsequence of this adenylation domains.

Activate Windows
Go to Settings to activate Windows



Current (Primitive) Scoring

#matches

In the current primitive scoring, we simply compute

Activate Windows

Go to Settings to activate Windows.

#matches – μ · #mismatches – σ · #indels

penalties for mismatches and insertion and deletions.

Activate Windows
Go to Settings to activate Windows.

#matches –
$$\mu$$
 · #mismatches – σ · #indels

score in our alignment game changes. Before it was 4, now it is -7.

Activate Windows

Go to Settings to activate Windows.

#matches –
$$\mu$$
 · #mismatches – σ · #indels

scoring matrix

In this case we essentially constructed the scoring matrix, which is a five by five

Activate Windows
Go to Settings to activate Windows

$$\#$$
matches – μ · $\#$ mismatches – σ · $\#$ indels

A C G T - A C G T -

A +1 -
$$\mu$$
 - μ - μ - σ A +1 -3 -5 -1 -3

C - μ +1 - μ - μ - σ C -4 +1 -3 -2 -3

G - μ - μ +1 - μ - σ G -9 -7 +1 -1 -3

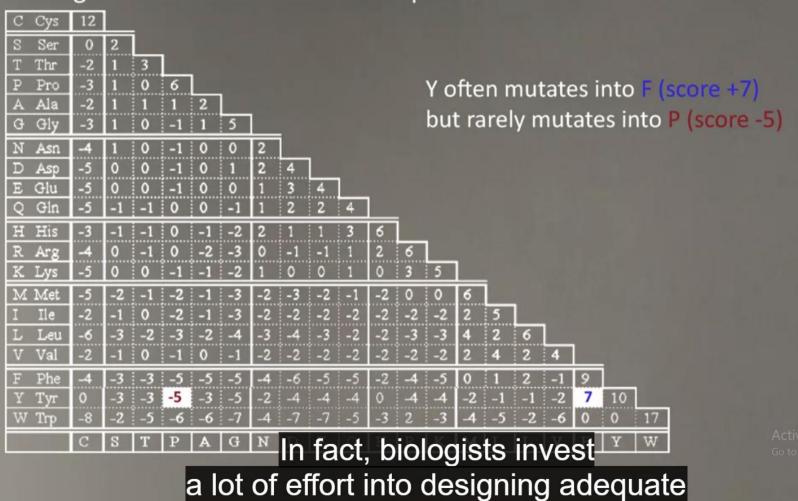
T - μ - μ - μ +1 - σ T -3 -5 -8 +1 -4

- - σ - σ - σ - σ - σ - - -4 -2 -2 -1

scoring matrix scoring matrix with arbitrary values

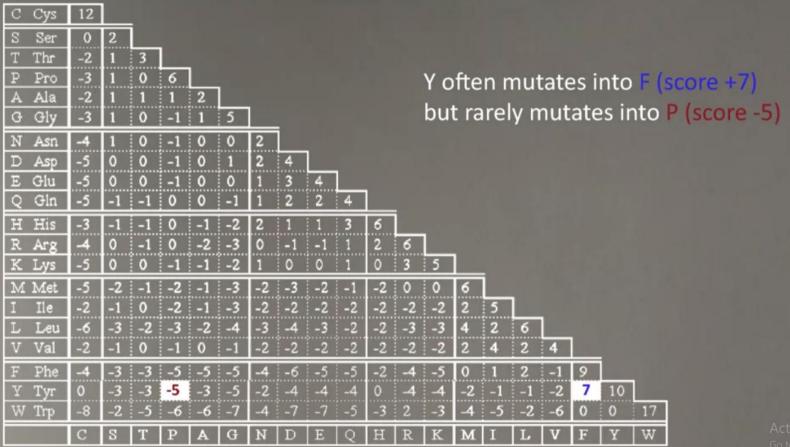
we can use it to play the alignment game.

Scoring Matrices for Amino Acid Sequences



Activate Windows Go to Settings to activate Wind

Scoring Matrices for Amino Acid Sequences

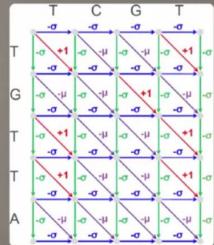


Activate Windows

Go to Settings to activate Window

Dynamic Programming Recurrence for the Alignment Graph

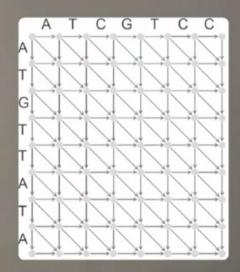
$$s_{i,j} = \max - \begin{cases} s_{i-1,j} - \sigma \\ s_{i,j-1} - \sigma \\ s_{i-1,j-1} + 1, \text{ if } v_i = w_j \text{ (match)} \\ s_{i-1,j-1} - \mu, \text{ if } v_i \neq w_j \text{ (mismatch)} \end{cases}$$





Dynamic Programming Recurrence for the Alignment Graph

$$s_{i,j} = \max - \begin{cases} s_{i-1,j} + score(v_i, -) \\ s_{i,j-1} + score(-, w_i) \\ s_{i-1,j-1} + score(v_i, w_j) \end{cases}$$

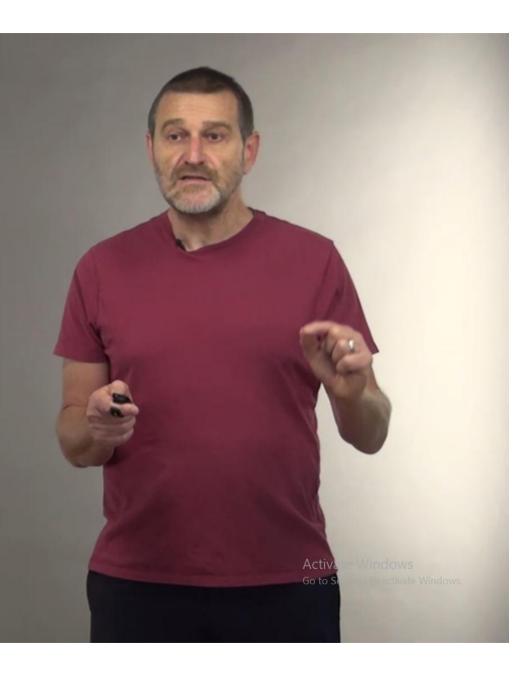




Global Alignment

Global Alignment Problem: Find the highest-scoring alignment between two strings by using a scoring matrix

- Input: Strings v and w as wellas a matrix score
- Output: An alignment of v and w whose alignment score (as defined by the scoring matrix score) is maximal among all possible alignments of v and w

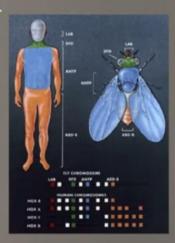


Homeobox Genes

Two genes in different species may be similar over short conserved regions and dissimilar over remaining regions

Homeobox genes have a short region called the **homeodomain** that is highly conserved among species

A global alignment may not find the homeodomain because it would try to align the *entire* sequence





Which Alignment is Better?

score =
$$17 \text{ (matches)} - 30 \text{ (indels)} = -13$$

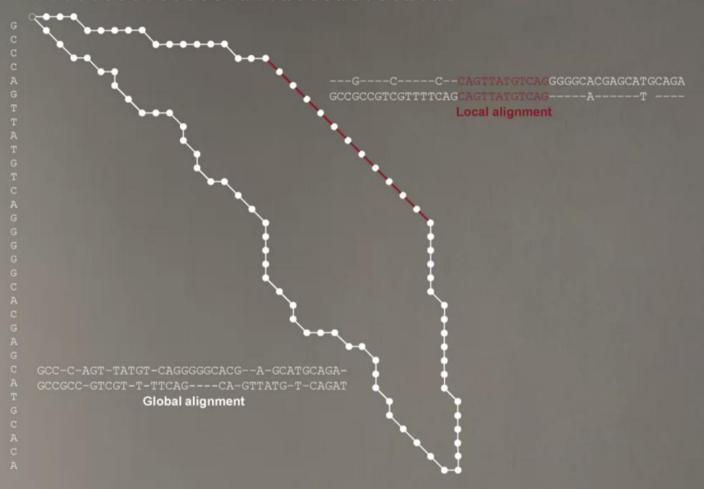
Which Alignment is Better?

score = 22 (matches) - 20 (indels)=2

GCC-C-AGT--TATGT-CAGGGGGCACG--A-GCATGCAGA-GCCGCC-GTCGT-T-TTCAG----CA-GTTATG--T-CAGAT

score = 17 (matches) - 30 (indels)=-13

---G----C--CAGTTATGTCAGGGGGCACGAGCATGCAGA
GCCGCCGTCGTTTTCAGCAGTTATGTCAG----A----T---local alignment



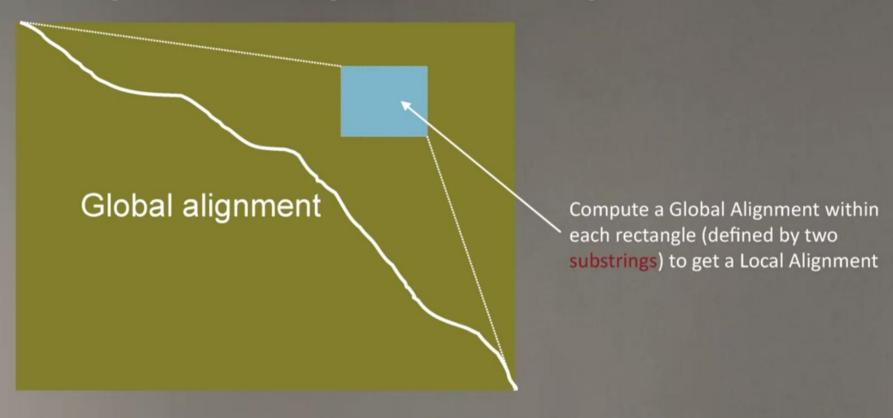
Activate Windows

To to Settings to activate Windows

Activate Windows

So to Settings to activate Windows

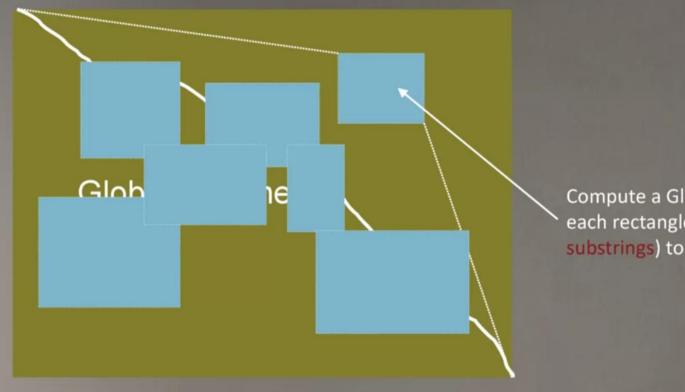
Local Alignment = Global Alignment in a Subrectangle



Activate Windows

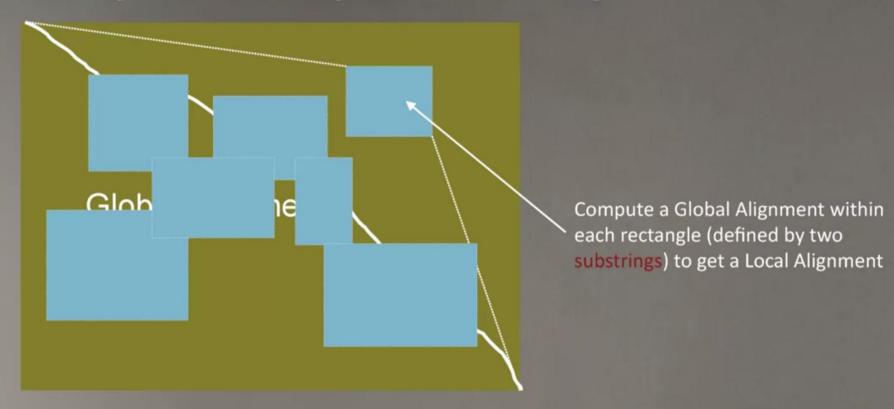
Go to Settings to activate Windows

Local Alignment = Global Alignment in a Subrectangle



Compute a Global Alignment within each rectangle (defined by two substrings) to get a Local Alignment

Local Alignment = Global Alignment in a Subrectangle



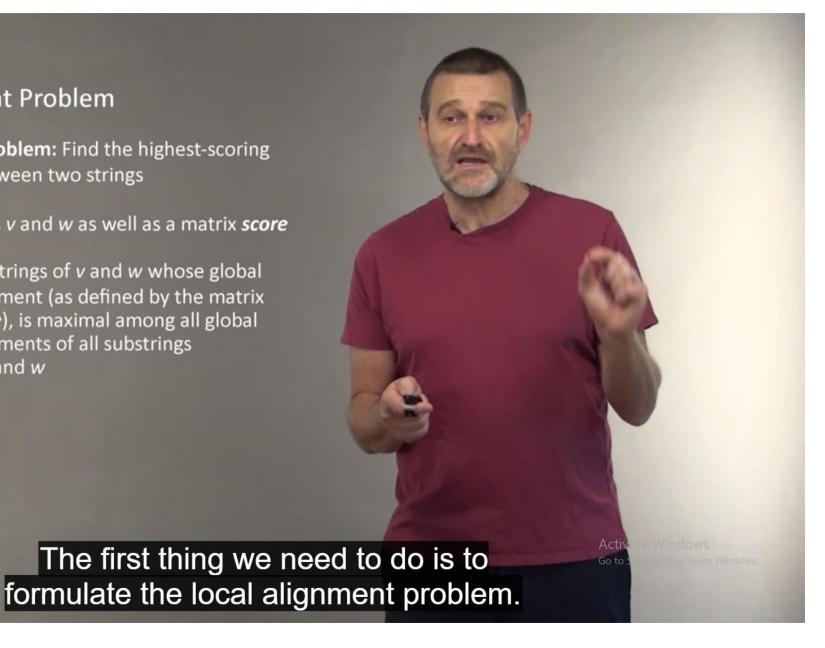
What can we do to come up with a practical

Activate Windows

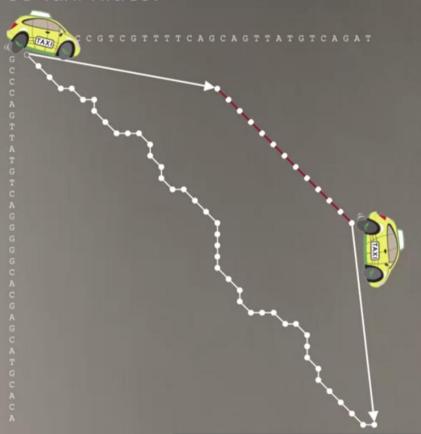


Local Alignment Problem: Find the highest-scoring local alignment between two strings

- Input: Strings v and w as well as a matrix score
- Output: Substrings of v and w whose global alignment (as defined by the matrix score), is maximal among all global alignments of all substrings of v and w



Free Taxi Rides!



GCC-C-AGT-TATGT-CAGGGGGCACG--A-GCATGCAGA-GCCGCC-GTCGT-T-TTCAG----CA-GTTATG-T-CAGAT

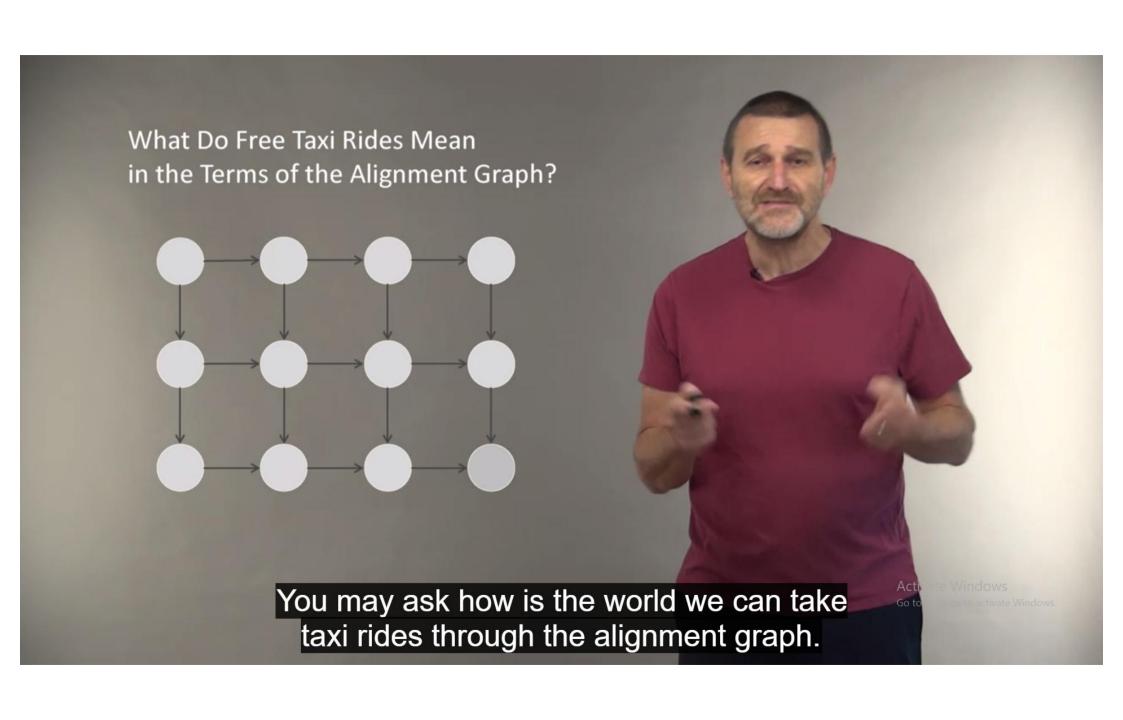
Global alignment

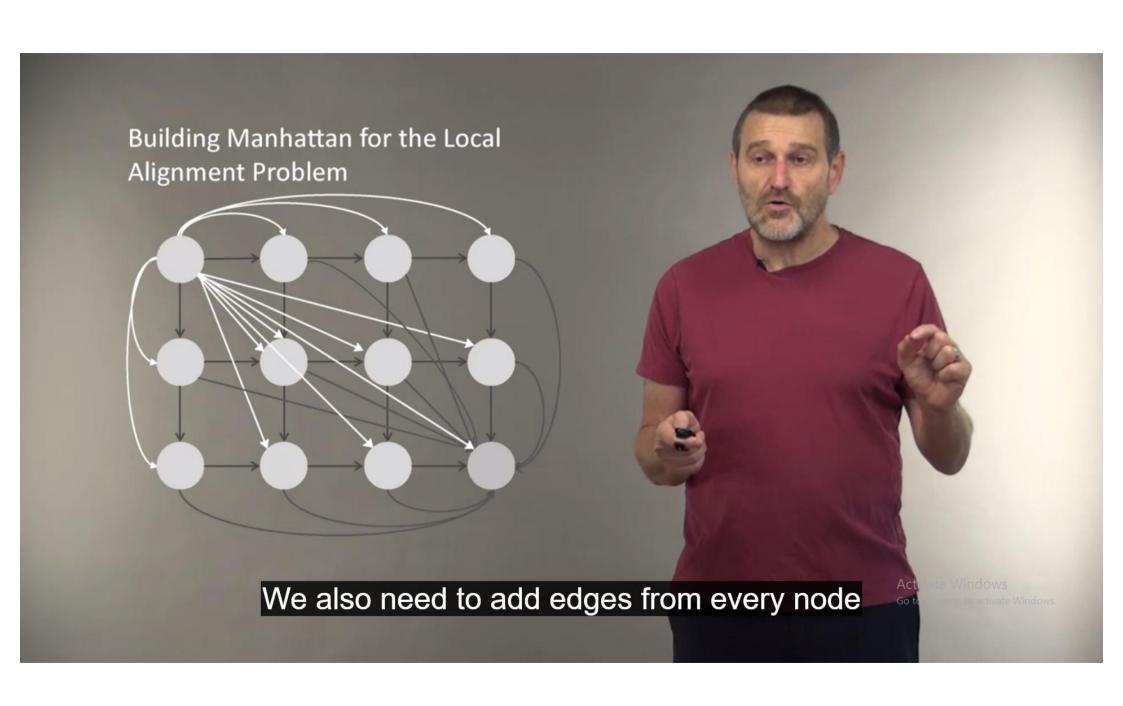
---G----C--CAGTTATGTCAGGGGGCACGAGCATGCAGA
GCCGCCGTCGTTTTCAGCAGTTATGTCAG----A----T
Local alignment

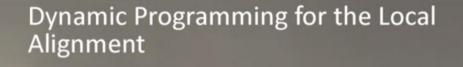
this problem: let's introduce free taxi rides through the alignment graph.

Activate Windows

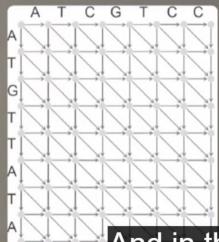
Go to Settings to activate Windows





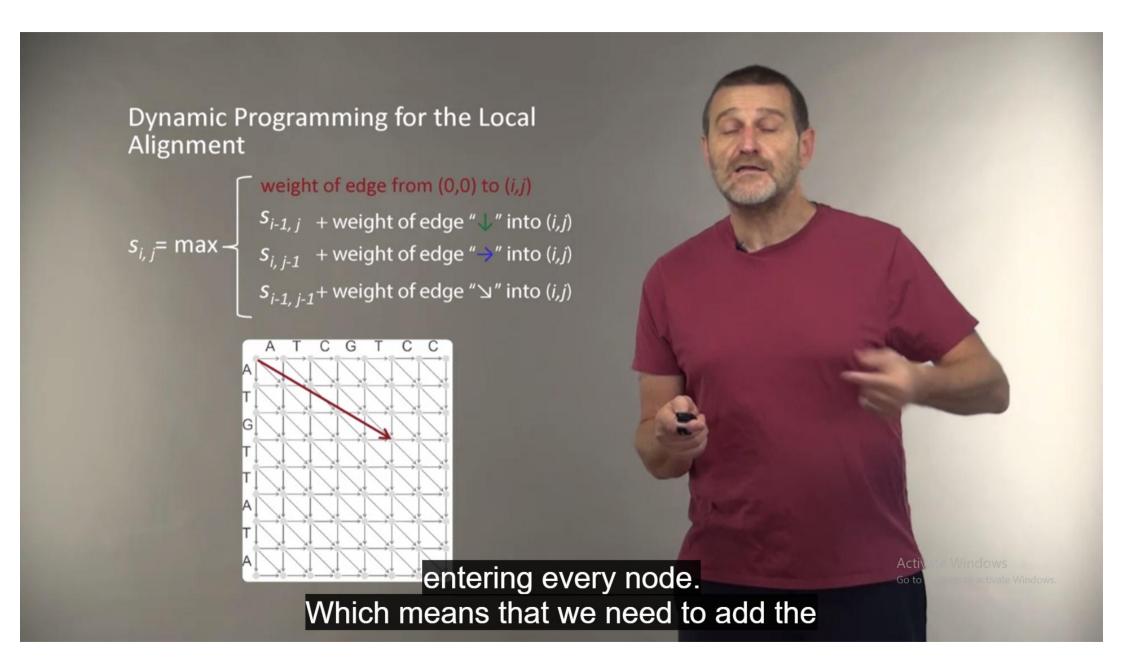


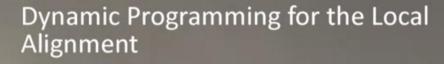
$$S_{i,j} = \max \begin{cases} S_{i-1,j} + \text{weight of edge "} \downarrow \text{" into } (i,j) \\ S_{i,j-1} + \text{weight of edge "} \rightarrow \text{" into } (i,j) \\ S_{i-1,j-1} + \text{weight of edge "} \searrow \text{" into } (i,j) \end{cases}$$



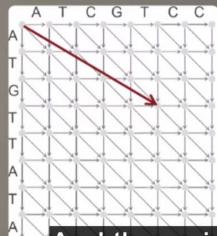
And in the end how our dynamic







$$S_{i,j} = \max \begin{cases} 0 \\ S_{i-1,j} + \text{weight of edge "} \downarrow \text{" into } (i,j) \\ S_{i,j-1} + \text{weight of edge "} \rightarrow \text{" into } (i,j) \\ S_{i-1,j-1} + \text{weight of edge "} \searrow \text{" into } (i,j) \end{cases}$$



And the weight of this edge since our taxi rides are free is zero.

