ENGG1811 Computing for Engineers

Week 11 Part A

Matlab: Hard computation problems

Algorithms

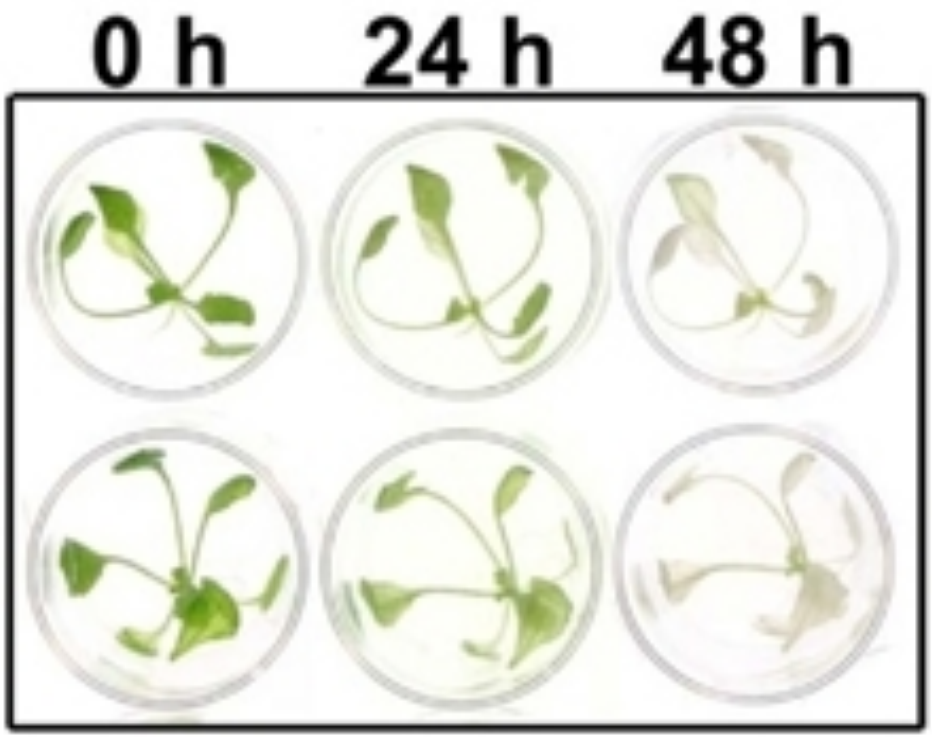
- Using algorithms to solve computation problems
- What you have done: Search for a number in a sorted list of numbers
  - Simple scan
  - Binary search

These are not the hardest type of computation problems!

- What are the hard and difficult computation problems?

These are not ordinary plants

- These plants turned white for a reason. Can you guess which substance made it turned white?



The technical paper

OPEN ACCESS Freely available online

PLoS one

Programmable Ligand Detection System in Plants through a Synthetic Signal Transduction Pathway

Mauricio S. Antunes[1☯], Kevin J. Morey[1☯], J. Jeff Smith[2¤a], Kirk D. Albrecht[1], Tessa A. Bowen[1], Jeffrey K. Zdunek[1], Jared F. Troupe[1], Matthew J. Cuneo[2¤b], Colleen T. Webb[1], Homme W. Hellinga[2], June I. Medford[1]*

1 Department of Biology, Colorado State University, Fort Collins, Colorado, United States of America, **2** Department of Biochemistry, Duke University Medical Center, Durham, North Carolina, United States of America

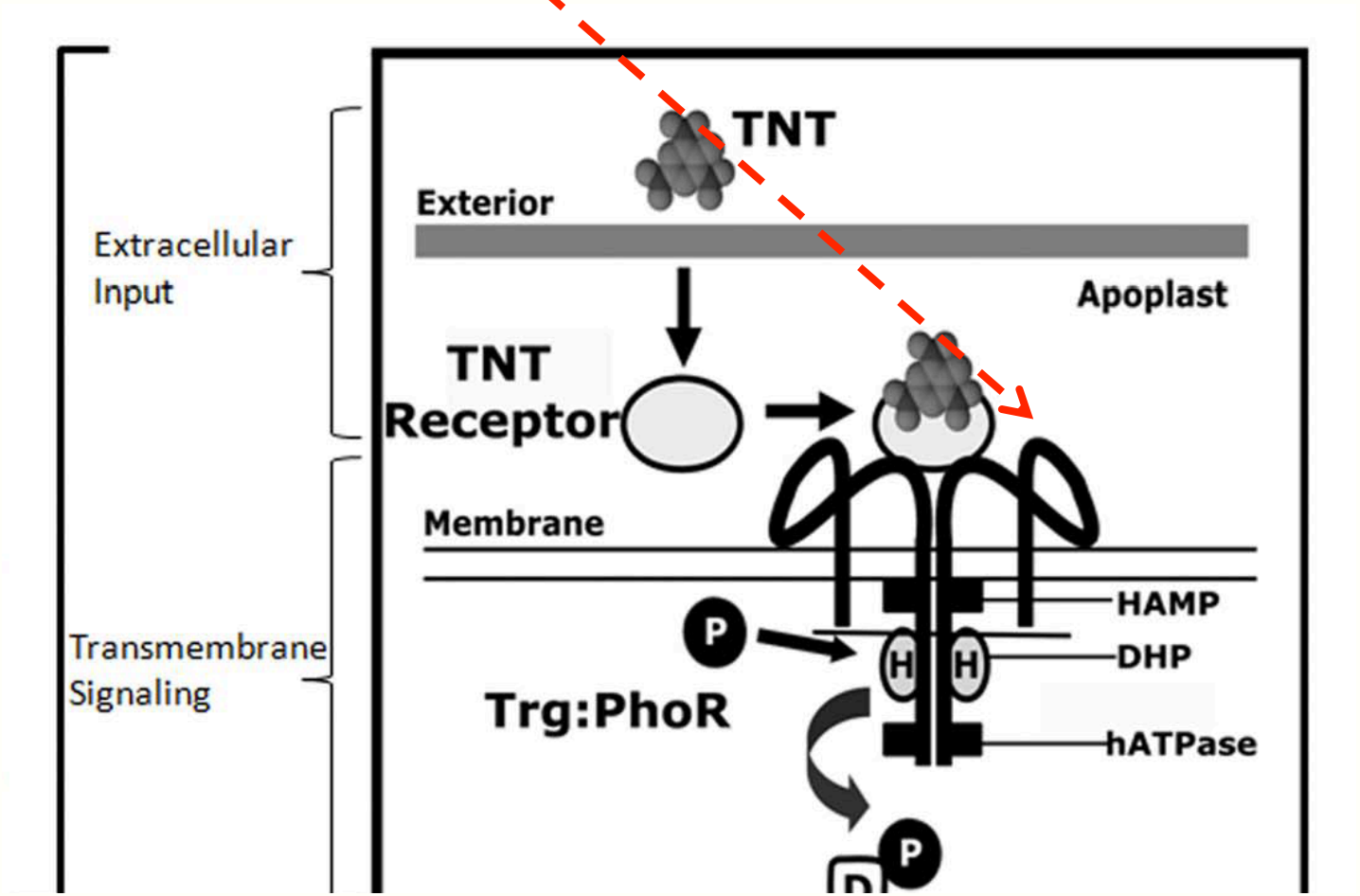
Background

There is an unmet need to monitor human and natural environments for substances that are intentionally or unintentionally introduced. A long-sought goal is to adapt plants to sense and respond to specific substances for use as environmental monitors. Computationally re-designed periplasmic binding proteins (PBPs) provide a means to design highly sensitive and specific ligand sensing capabilities in receptors. Input from these proteins can be linked to gene

http://wp.natsci.colostate.edu/medfordlab/
http://www.plosone.org/
http://www.wired.co.uk/news/archive/2011-01/28/bomb-detecting-plant

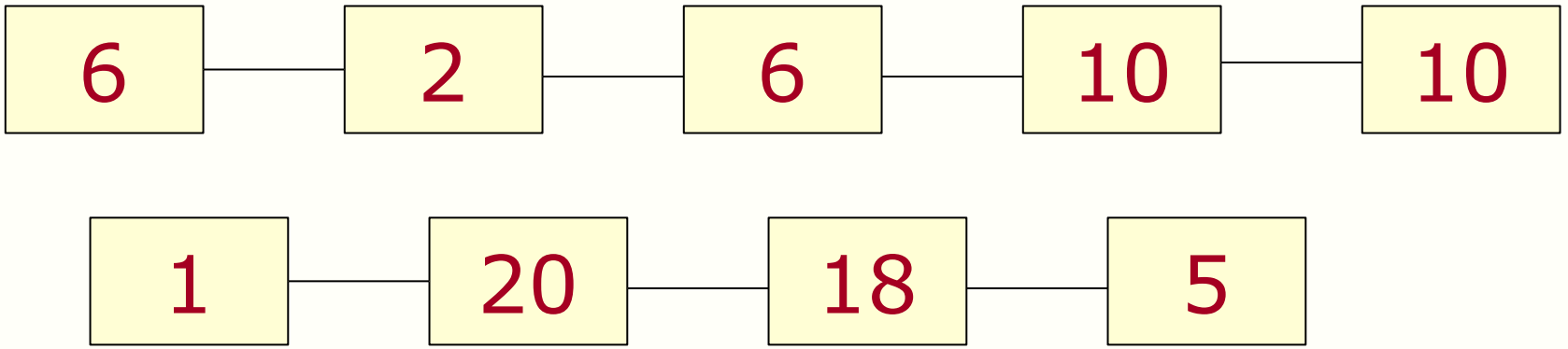
The design problem

- Find a protein that matches the shape of a TNT molecule



Protein

- Protein is a chain of amino acids
- There are 20 amino acids
- For simplification, we think about protein as a sequence of numbers
  - We only use numbers 1-20
  - Any number can be at any position
- Question: How many different proteins, which have 4 numbers, can you have?



Designing protein

- You want to design a protein which is 100 amino acid long
- Don't know which one is good, so you need to check each one of them out

- The number of possibilities is
  -
- This is possibilities

- Question: You have 1000 computers, each computer can check one possibility in 10^(-6) second, how long does it take to check all the possibilities?

Hard computation problems

- Many **hard** computation problems involve looking for the best solution within an enormous number of possibilities

- Other examples:
  - Scheduling of fleet, production etc.
  - University time tabling
  - Sensor placement for monitoring water networks (Week 3)
    - We might have tricked you into believing the problem was quite solvable. We had only 5 possible locations, but real problems have > 100 locations, which means > 10^30 possibilities.

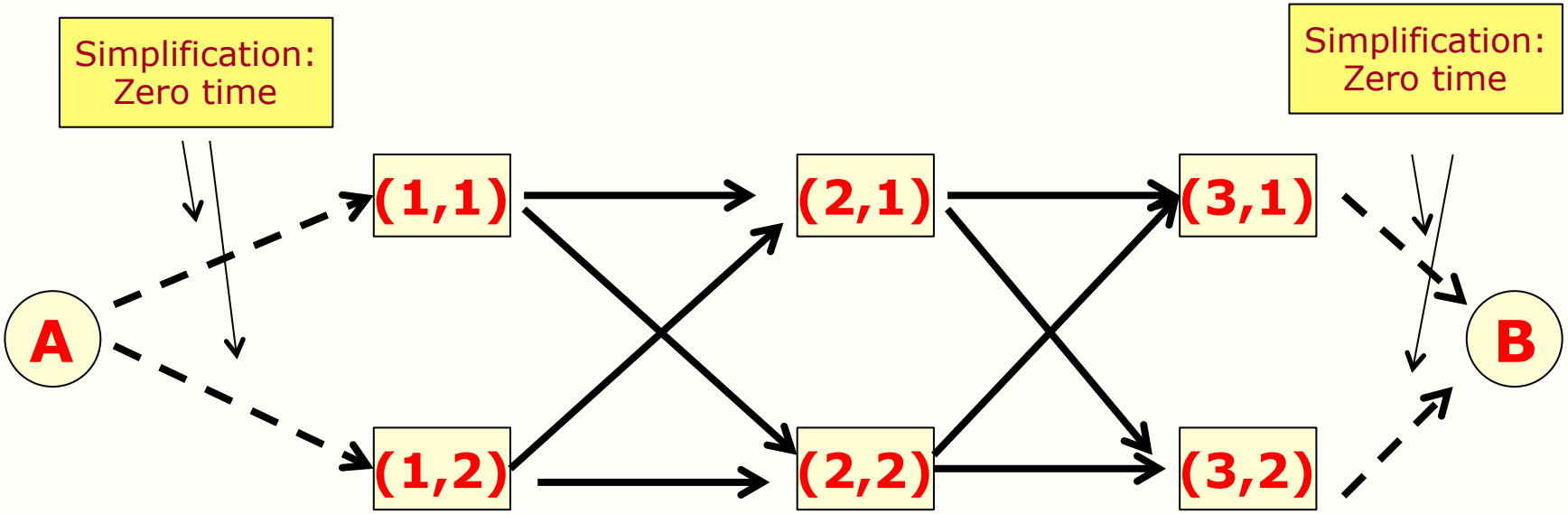
What can you do with hard computation problems?

Smart algorithms

- The protein design problem is solved by reducing the number of possibilities that we have to check. Unfortunately, the algorithm is complicated.
- Instead, we will look at another algorithm that tries to keep the number of possibilities that we have to check low
- The algorithm is called Vitebri algorithm after its inventor
- The algorithm has many applications
  - All mobile phones use this algorithm
  - Automatic speed recognition
  - Computational biology

Problem definition

- Imagine you want to go from a city A and a city B
- There are many choices of road in between
- Each road takes a certain amount of time
- You want to go from A to B in the shortest time possible
- City (k,i) = city i at stage k

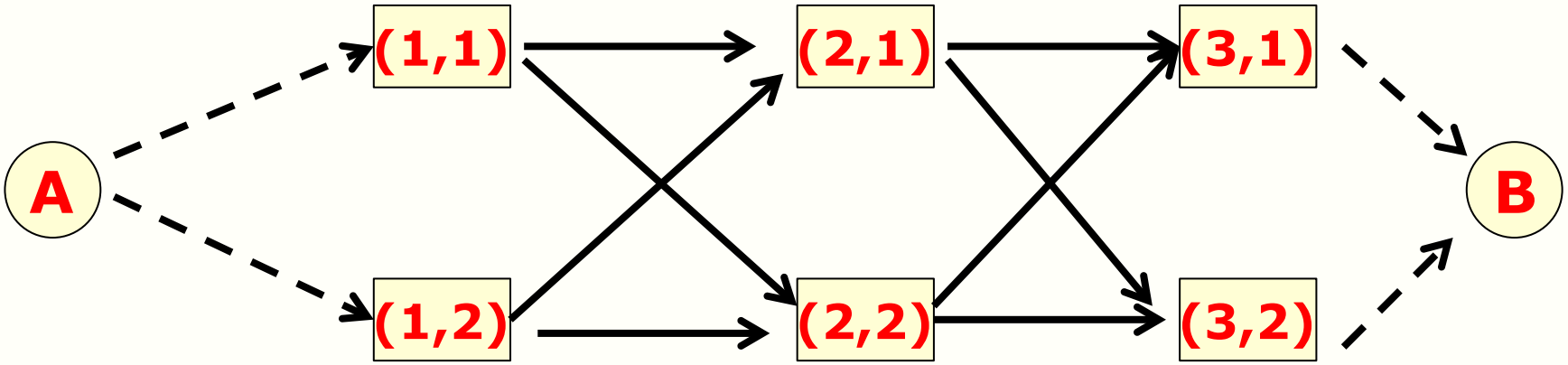


Path time (example 1)

From \ To	(2,1)	(2,2)
(1,1)	5	4
(1,2)	2	3

From \ To	(3,1)	(3,2)
(2,1)	7	1
(2,2)	8	9

- Time from A to B through (1,1) – (2,1) – (3,1)
  = 5 + 7 = 12

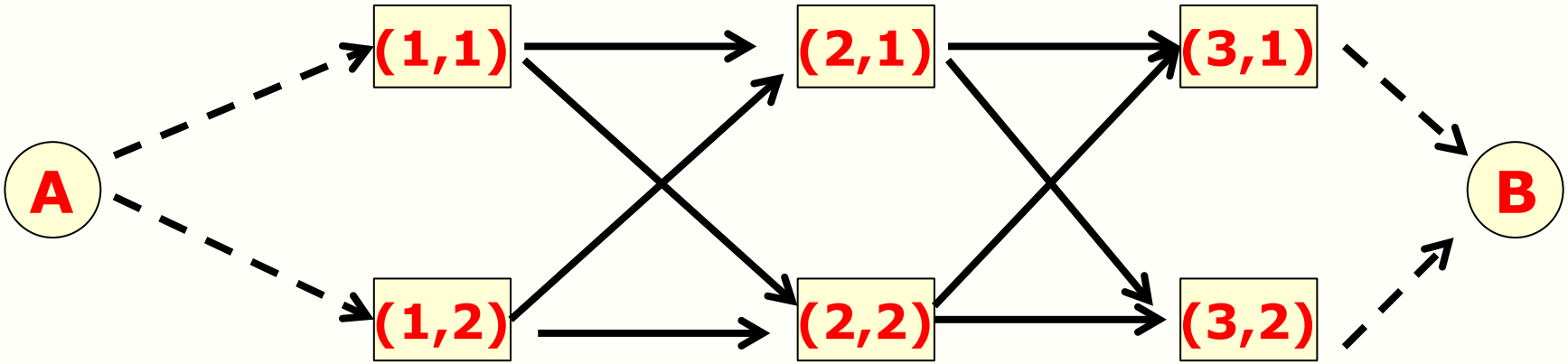


Path time (example 2)

From \ To	(2,1)	(2,2)
(1,1)	5	4
(1,2)	2	3

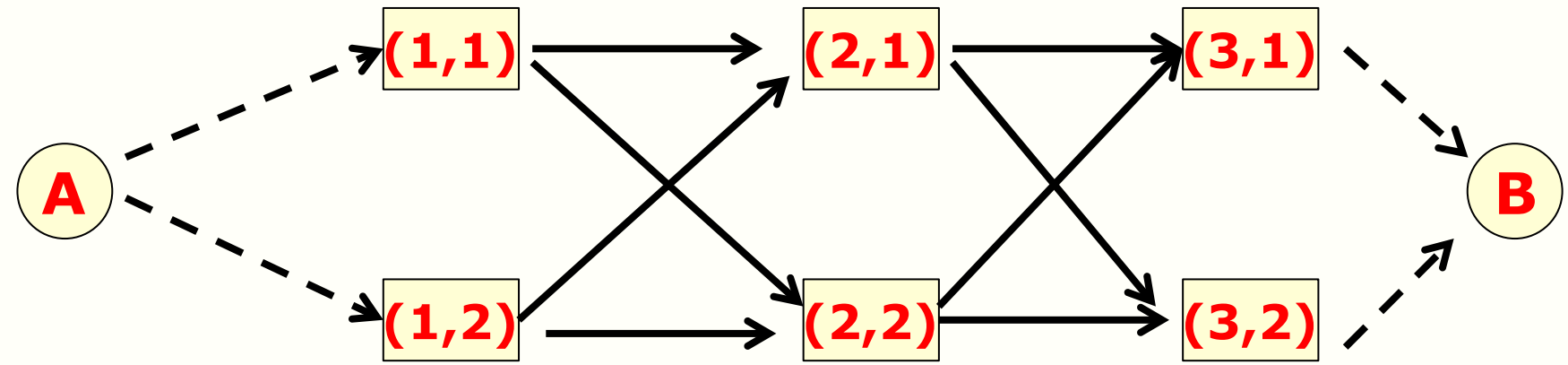
From \ To	(3,1)	(3,2)
(2,1)	7	1
(2,2)	8	9

- Time from A to B through (1,2) – (2,1) – (3,2)
  = 2 + 1 = 3



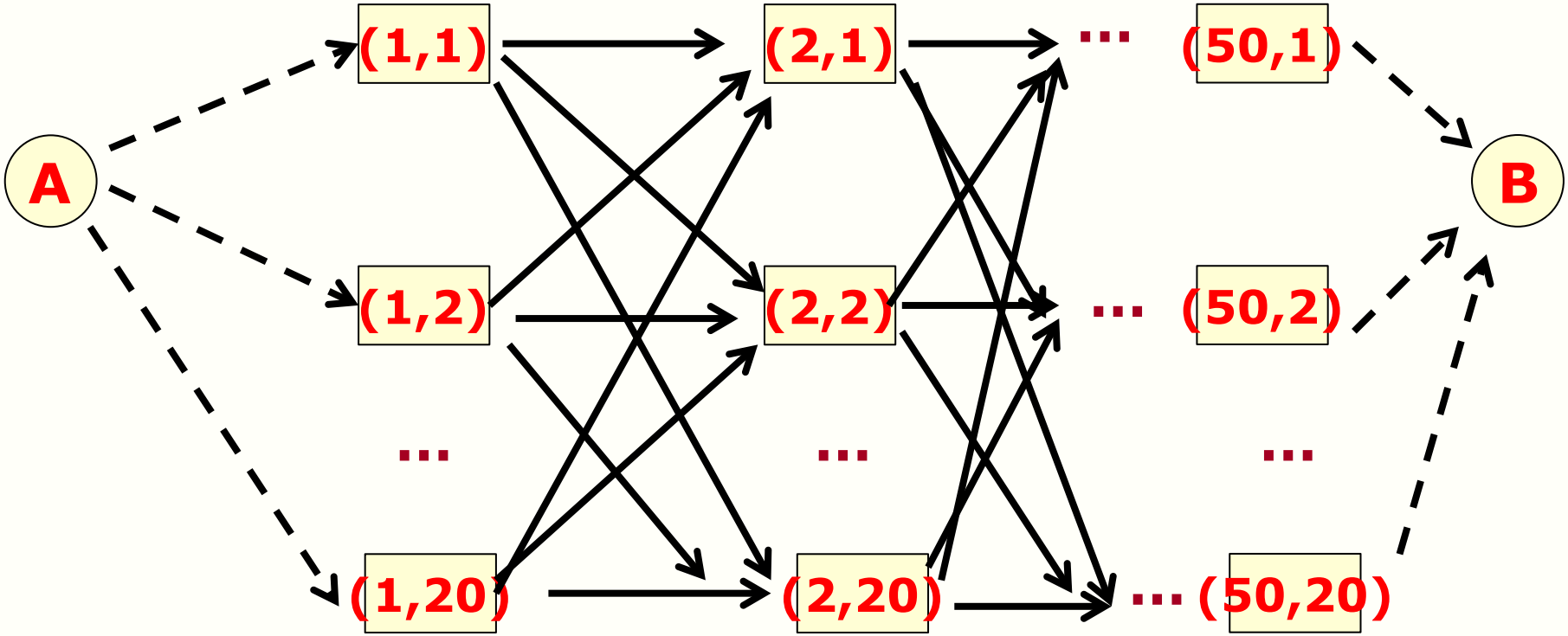
Enumerating all possible paths

- How many possible paths are there from A and B?



How many paths?

- How many possible paths are there from A and B?
  - 50 stages, 20 cities per stage
  - Answer:



Exhaustive search

- The steps are
  1. Enumerate all possible paths
  2. For each path, find the travel time
  3. Find the path with the minimum time

- Exhaustive search not possible for reasonable problem size

- Need a smarter algorithm

Question

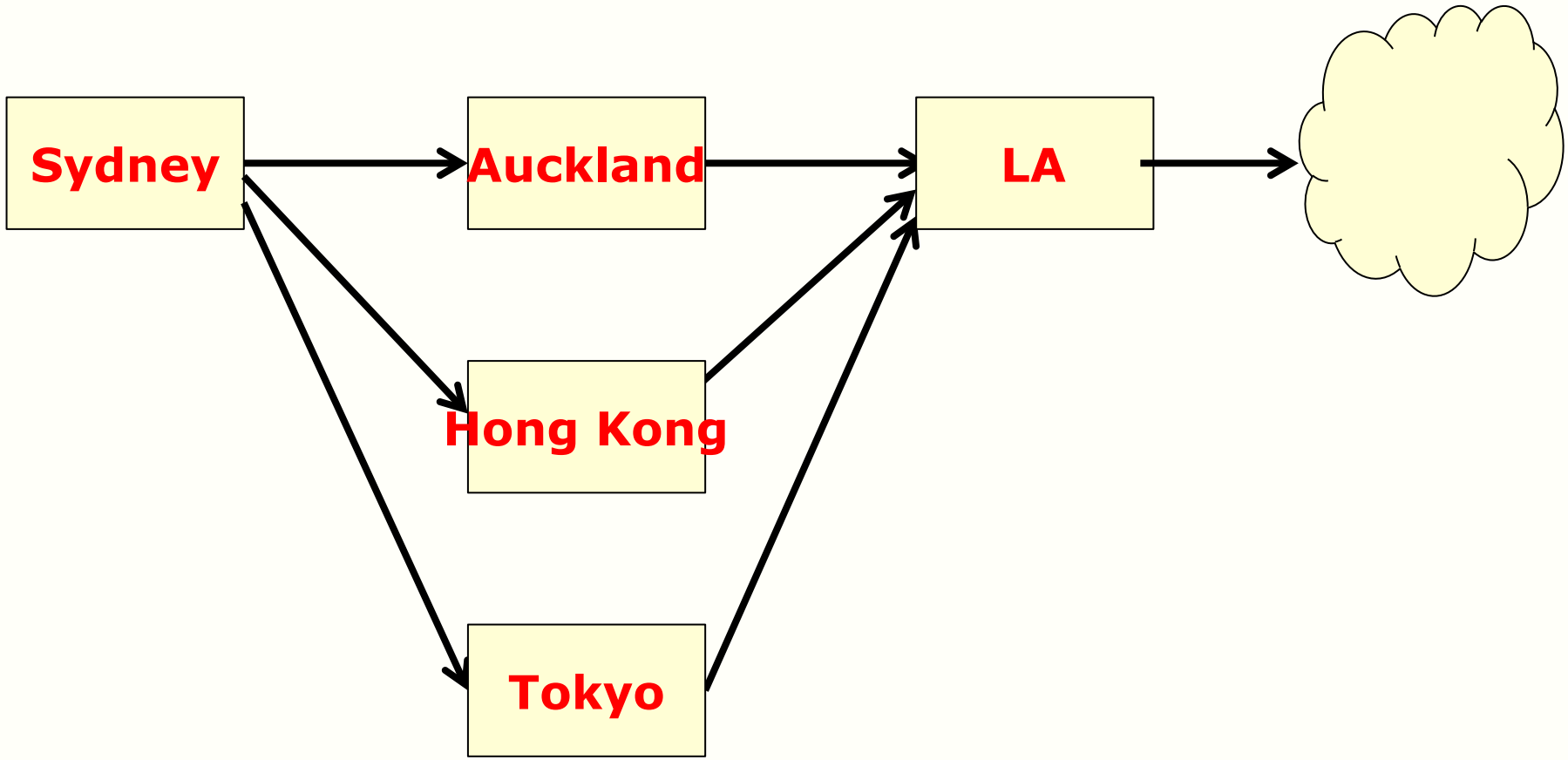
- You want to go from Sydney to Montreal in the shortest time possible
- You've made up your mind to go via LA
- You don't know how you will go from LA to Montreal
- There are no direct flights from Sydney to LA (sorry, all full) but the options are:
  - Via Auckland, 47 hours
  - Via Hong Kong, 31 hours
  - Via Tokyo, 36 hours
- For simplification, we assume all three plans land at LA at the same time?
- Which flight will you take to LA to minimise the overall travel time to Montreal? Can you decide even you don't know how you are going to travel from LA to Montreal?

Question (cont'd)

- To help you to answer this question, we will fill in the following table for the total travel time from Sydney to Montreal assuming 3 different options of travelling from LA to Montreal
  - Via Chicago, 20 hours
  - Via Boston, 12 hours
  - Via "don't know yet", x hours

To LA \ From LA	Via Chicago	Via Boston	Via don't know yet
Via Auckland			
Via Hong Kong			
Via Tokyo			

The problem that you had just solve can be depicted as

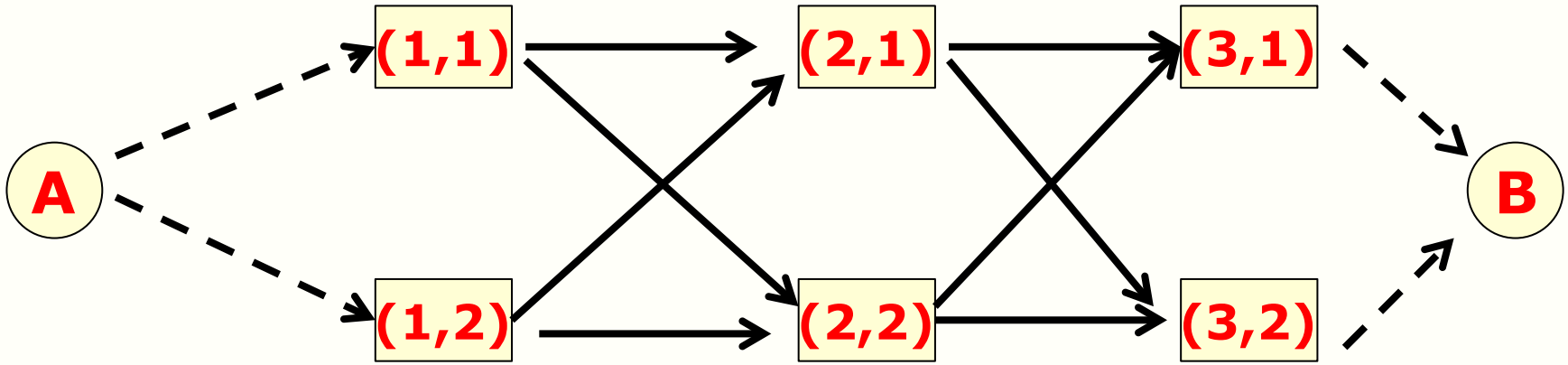


Using the observation

From \ To	(2,1)	(2,2)
(1,1)	5	4
(1,2)	2	3

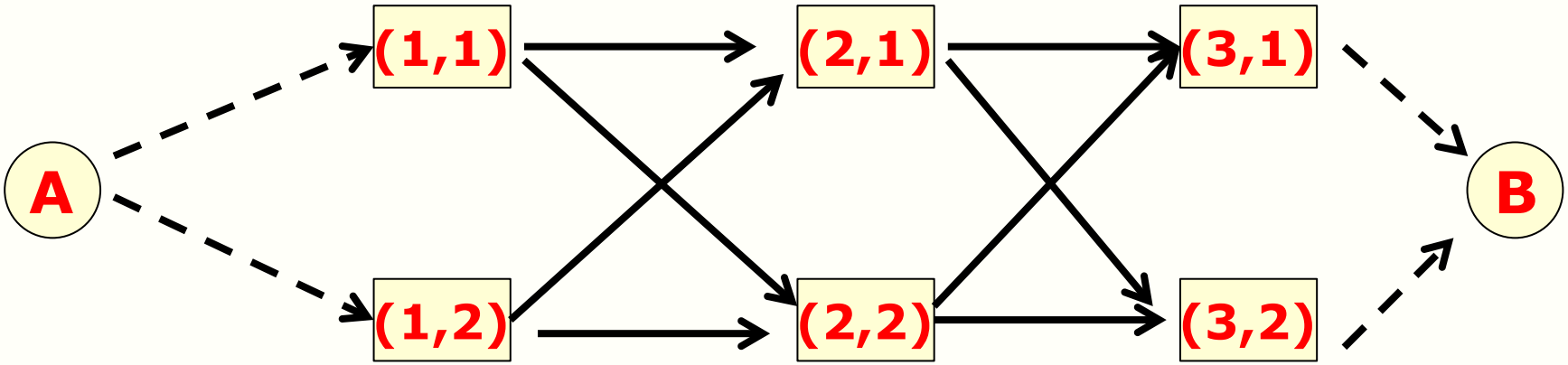
From \ To	(3,1)	(3,2)
(2,1)	7	1
(2,2)	8	9

- There are two paths going through (2,2)
  - (1,1) – (2,2)
  - (1,2) – (2,2)



Problem parameters (1)

```
% Define the problem parameters
nStages = 5;            % Number of stages
nCitiesPerStage = 2;  % Number of cities per stage, CONSTANT
```



- Note: The code can be modified to work with any number of cities per stage, but for simplicity, we don't do it.

Problem parameters (2)

- Use random numbers as travel times

```
% Generate random travel times for each link
linkTime = rand(nCitiesPerStage,nCitiesPerStage,nStages);
```

- 3 dimensional array
- linkTime(i,j,k) = travel time from (k,i) to (k+1,j)

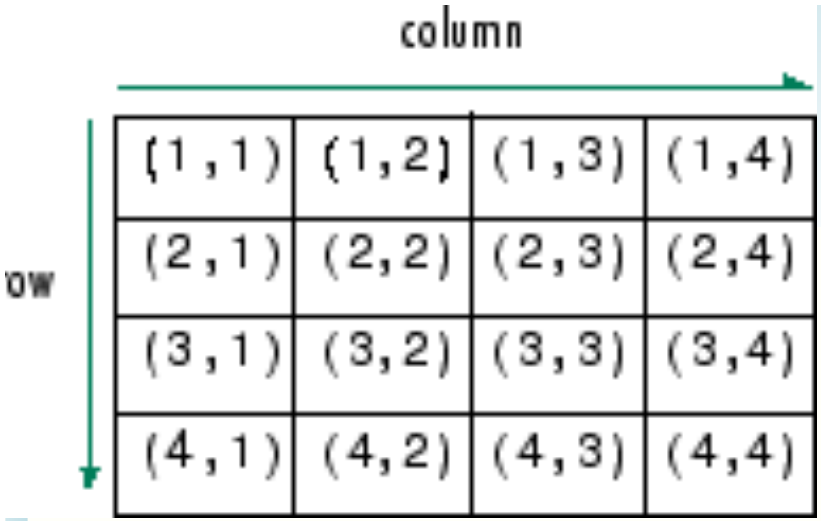
k = 1

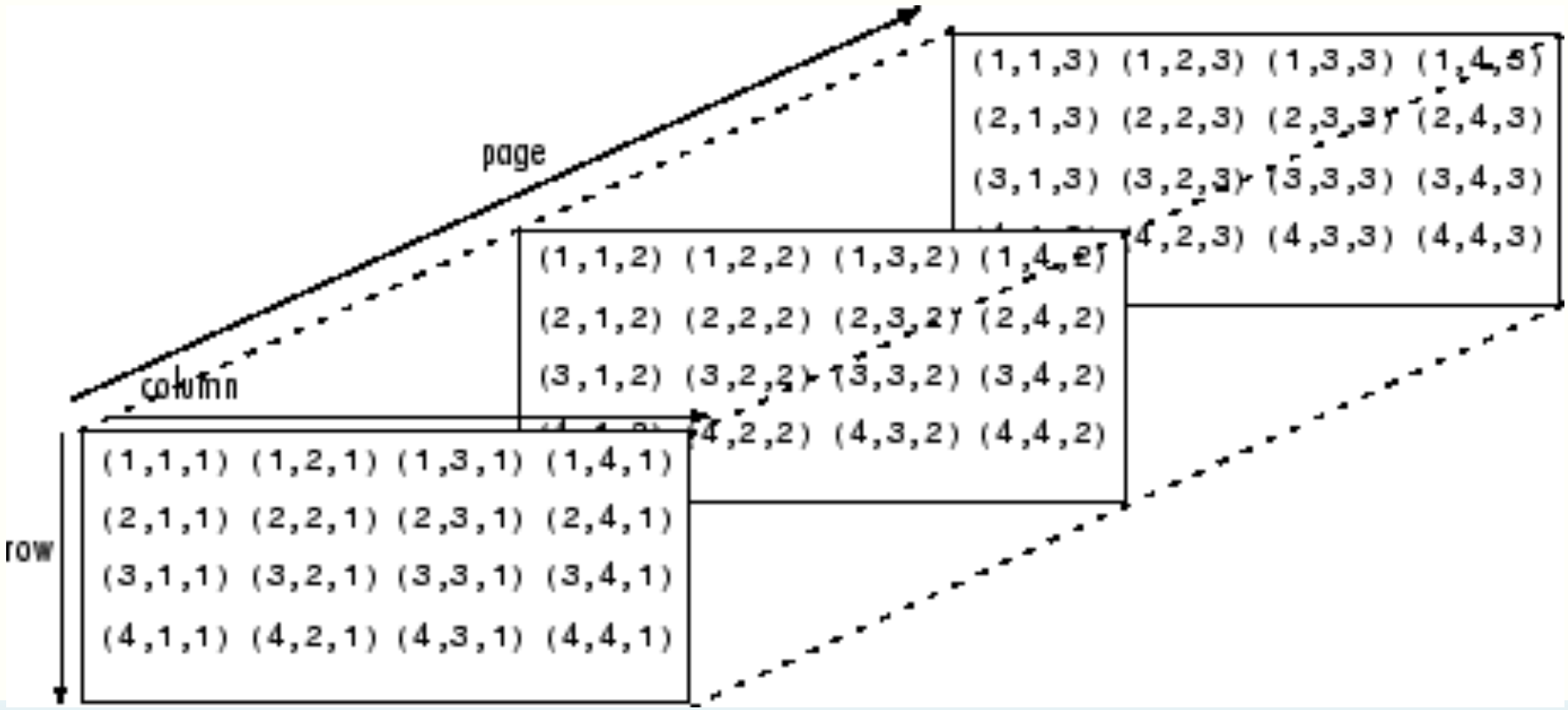
From \ To	(2,1)	(2,2)
(1,1)	5	4
(1,2)	2	3

k = 2

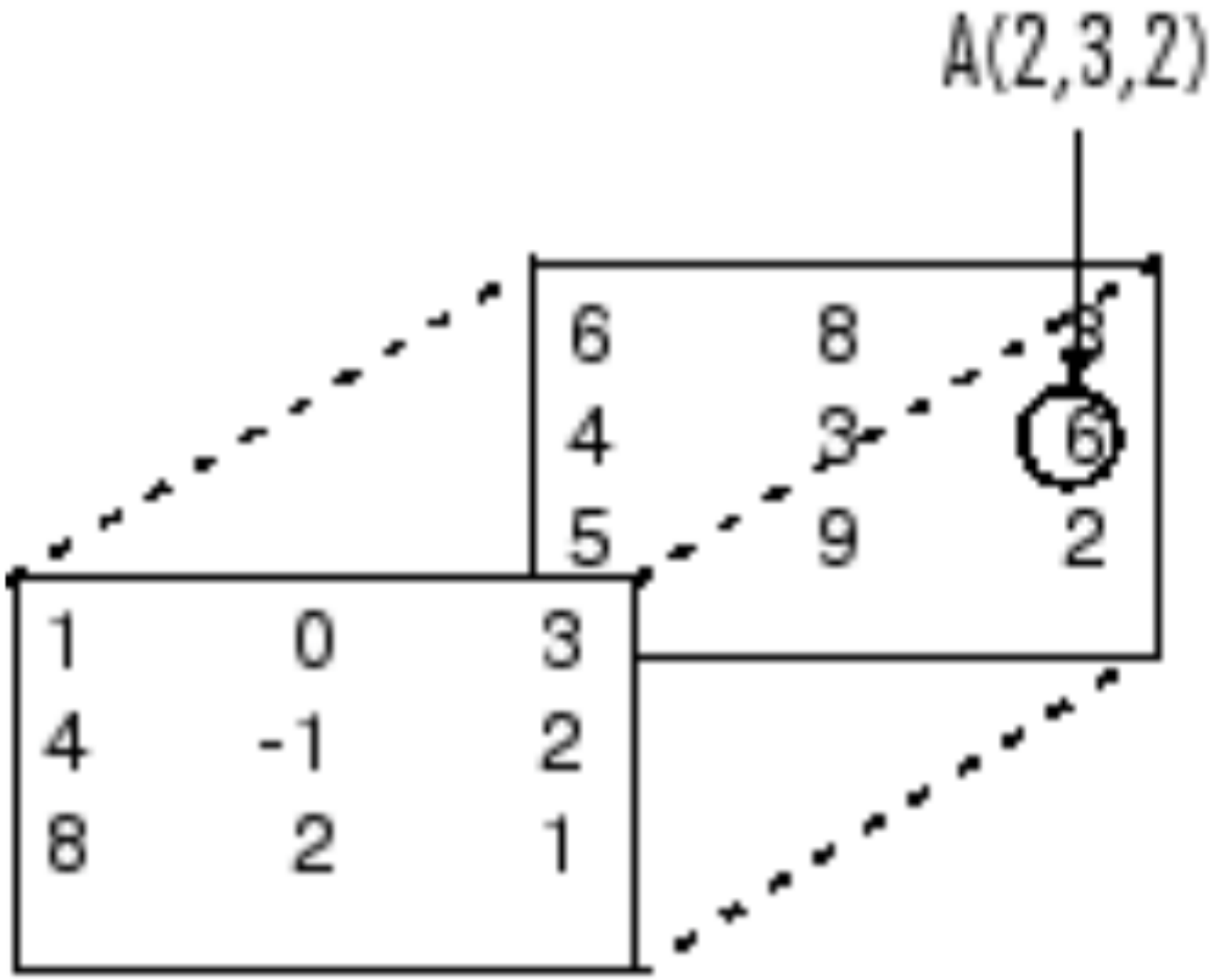
From \ To	(3,1)	(3,2)
(2,1)	7	1
(2,2)	8	9

Multi-dimensional array



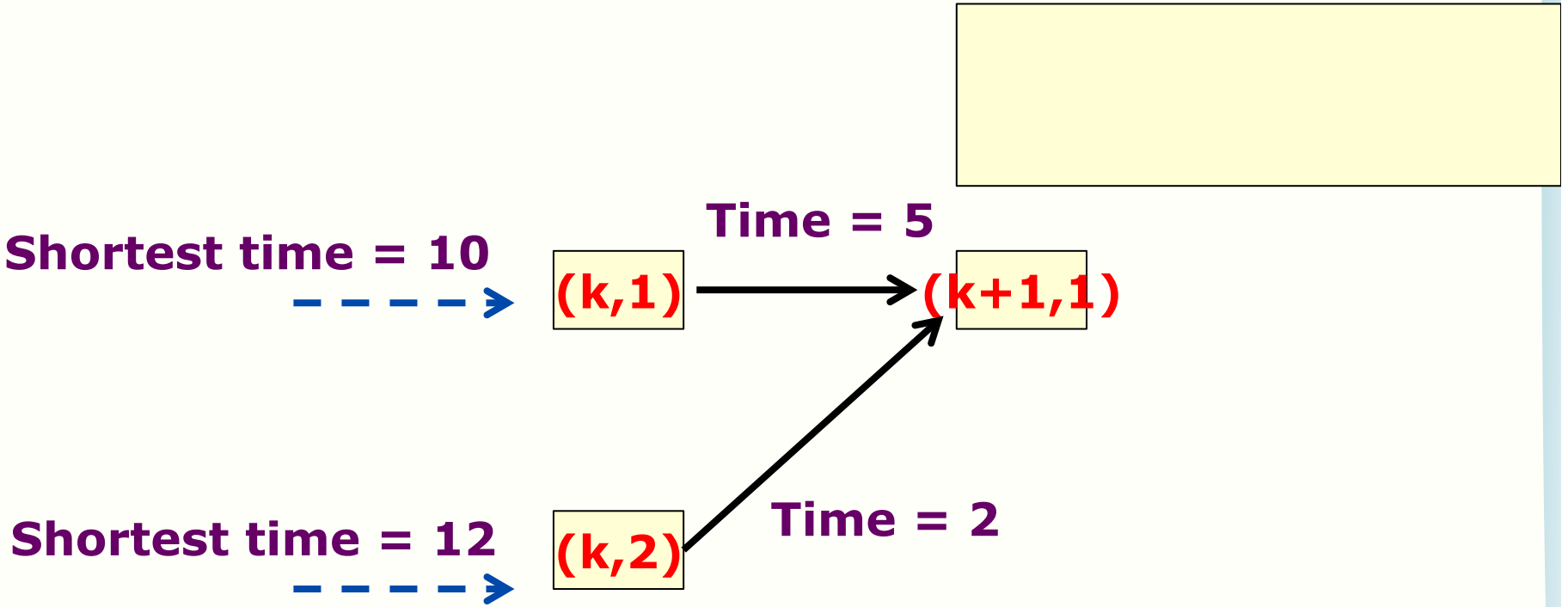


Multi-dimensional array (Example)



Question to help you to set up the iteration

- Assume you know
  - the shortest time to reach the cities in stage k
  - The time to go from all cities at stage k to stage (k+1)
- Question: For the example below, what is the shortest time to get to city (k+1,1) and which city should you use?



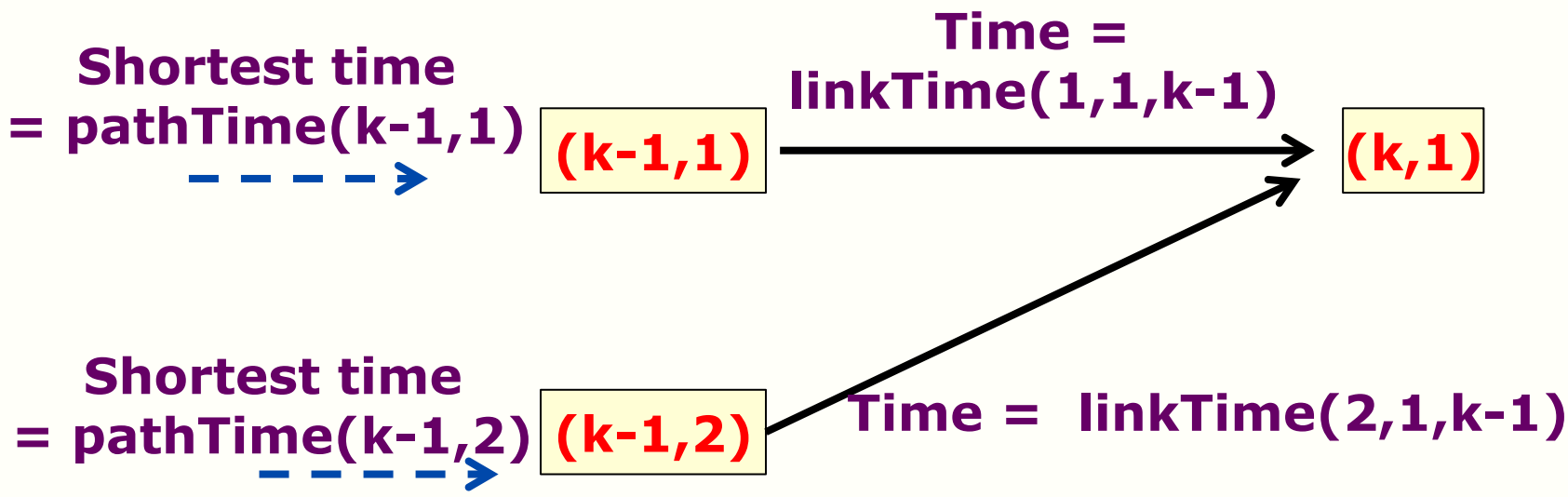
Data structure

```
% Define data structures
pathTime = zeros(nStages,nCitiesPerStage);
previousHop = zeros(nStages,nCitiesPerStage);
```

- pathTime(k,i)
  - The shortest travel time to get to city (k,i)
- previousHop(k,i)
  - The city to reach (k,i) from stage k-1 with the shortest path time

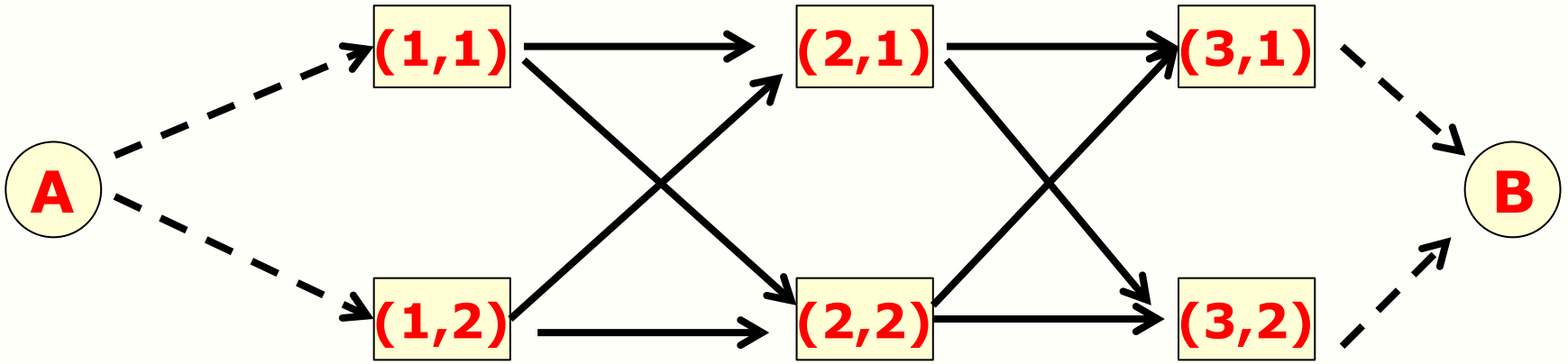
Iteration at stage k

```
% To Node 1 at Stage k
   timeFromNode1 = pathTime(k-1,1) + linkTime(1,1,k-1);
   timeFromNode2 = pathTime(k-1,2) + linkTime(2,1,k-1);
   if timeFromNode2 >= timeFromNode1
       pathTime(k,1) = timeFromNode1;
       previousHop(k,1) = 1;
   else
       pathTime(k,1) = timeFromNode2;
       previousHop(k,1) = 2;
   end
```



Output the path

- We have all previousHop(k,i)
  - The city to reach (k,i) from stage k-1 with the shortest path time
- Let us assume (3,2) is the shortest path to reach B
  - If previousHop(3,2) = 1, means we should use (2,1)
  - If previousHop(2,2) = 2, means we should use (1,2)



The whole program

- The whole program is in viterbi.m
- The second half of the file contains the code which uses exhaustive search to solve the same problem

- Why did I include the code for exhaustive search?
- Answer: For verification using two **independent** methods
- This is a way to check whether your program/algorithm is correct
- In my research, I come out with formulas/algorithms that no one has come out before. How do I verify that I have done something right? I try to get the same results using two independent methods.

Summary

- Difficult computation problems often require finding the best solution out of many possibilities
- I hope you can recognise these types of difficult problems in your own domain