{EPITECH. TECHNOLOGY}

B-CPE-200 : Lem-in

Projects

B-CPE-200 - Projects

B-CPE-200 : 20/02/23 - 04/06/23 **: 9 crédits**

Dante

Maze generation and resolution

1 – 2p
3 weeks

Lemin

Pathfinding and optimization algorithm

3 – 4p
4 weeks

Corewar

Simulation of combat in memory via a virtual machine

3 – 4p
5 weeks

Each project assesses these skills:

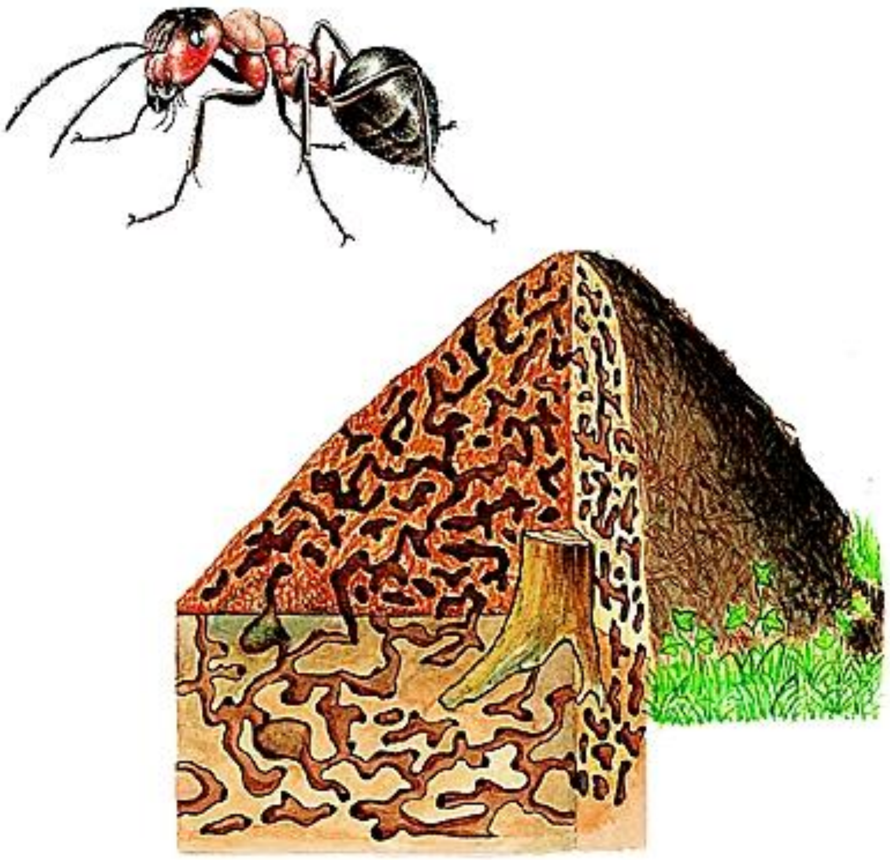
Algorithm | **Data structure** | **Parsing** | **Optimization** | **Robustness**

Lem-in

B-CPE-200 – Lem-in

Lem-in

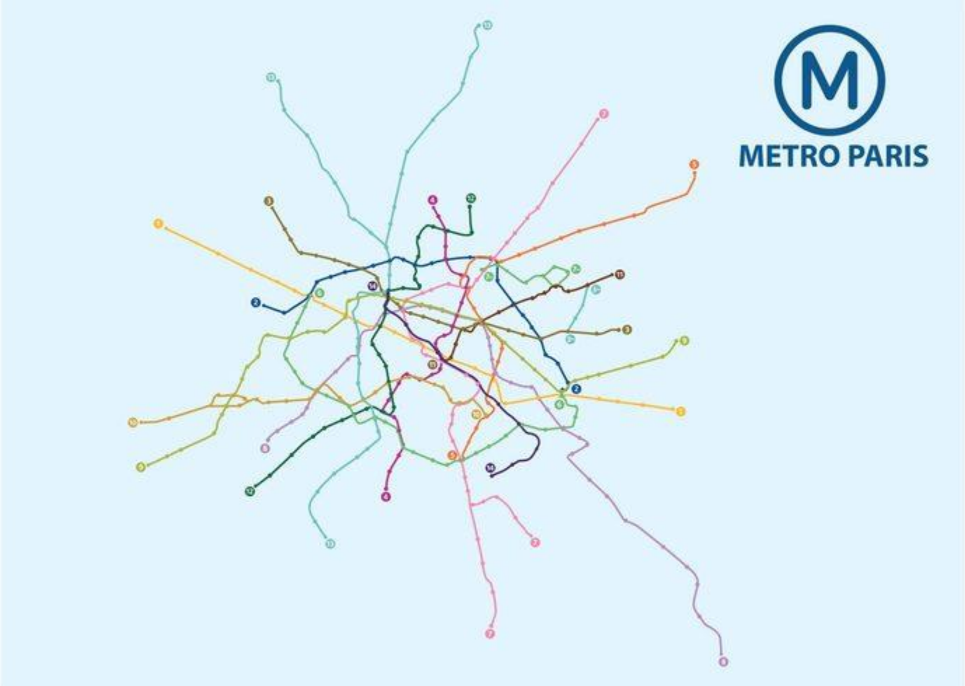
The goal of Lem-in is to work on pathfinding algorithms. The goal is not necessarily to manage the distance but rather the simultaneous occupation of the rooms.



B-CPE-200 – Pathfinding

Why ?

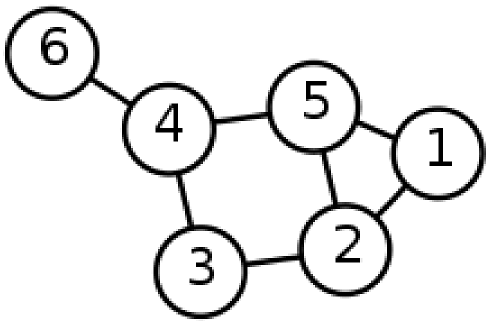
- Usefull for many projects (who said moves in video games ?)
- Pathfinding is everywhere (google maps, waze, subway network)



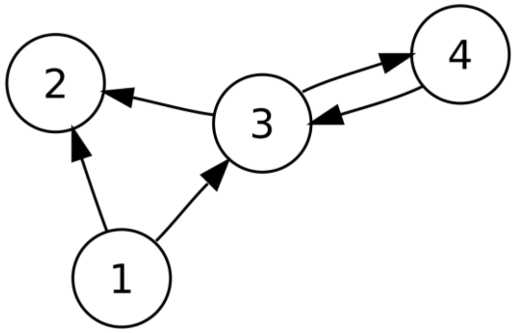
B-CPE-200 – Graphs

How to

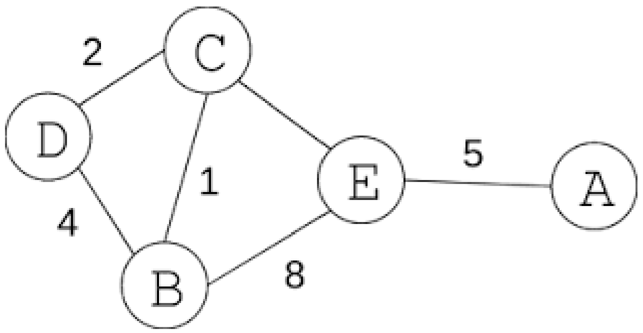
The solution of this kind of problem is feasible by applying graph theory



Graph

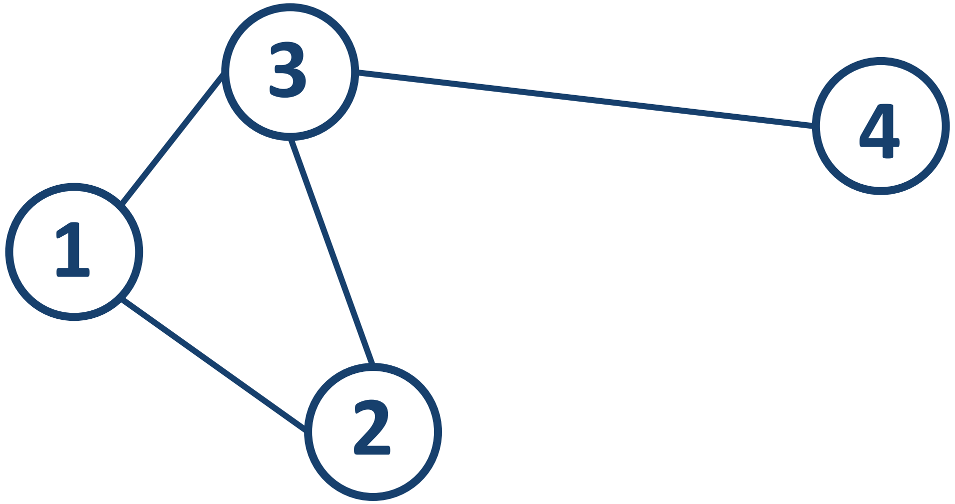


Oriented Graph



Weighted graph

B-CPE-200 – Graphs

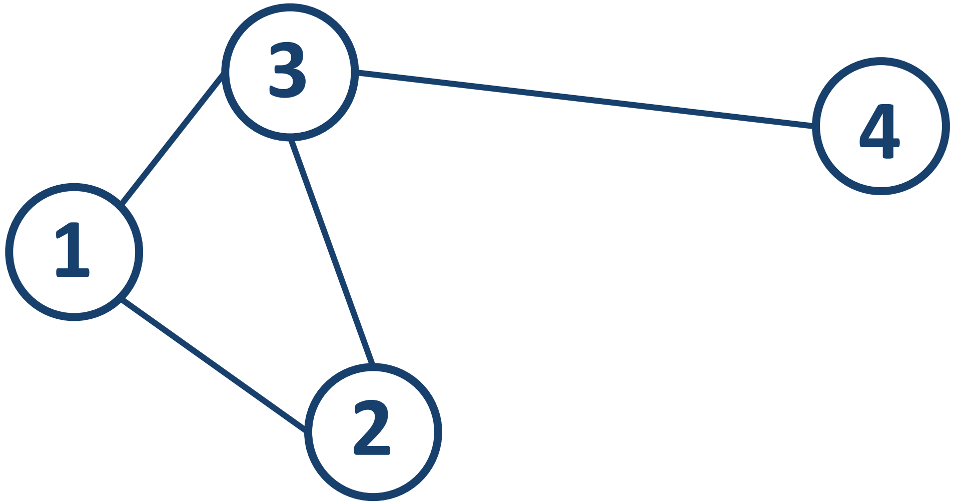


There are 2 ways to represent our graph in C :

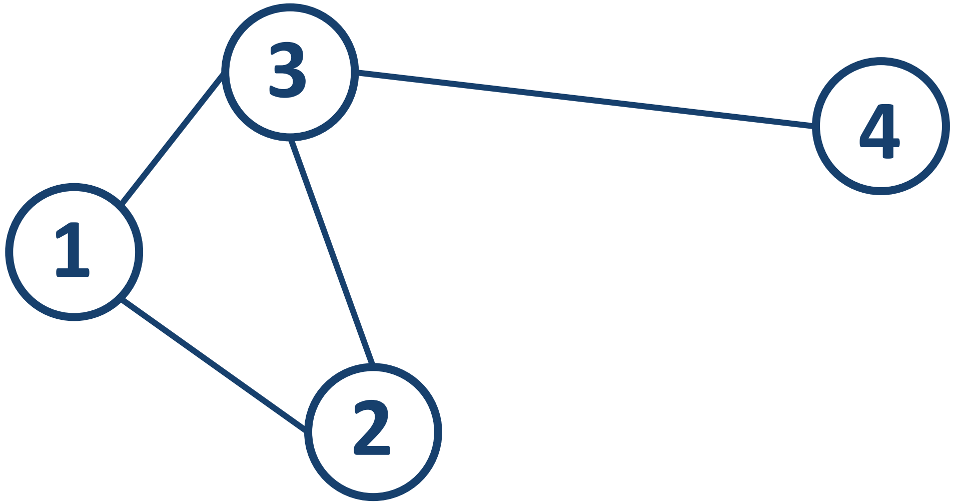
Linked lists

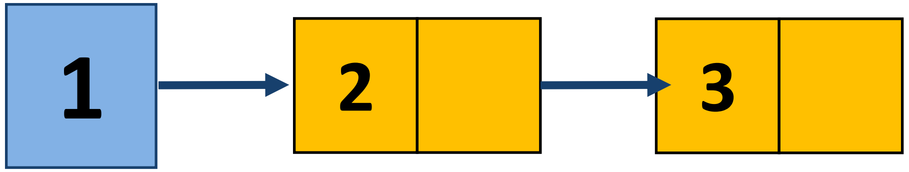
Adjacency matrix

B-CPE-200 – Linked lists

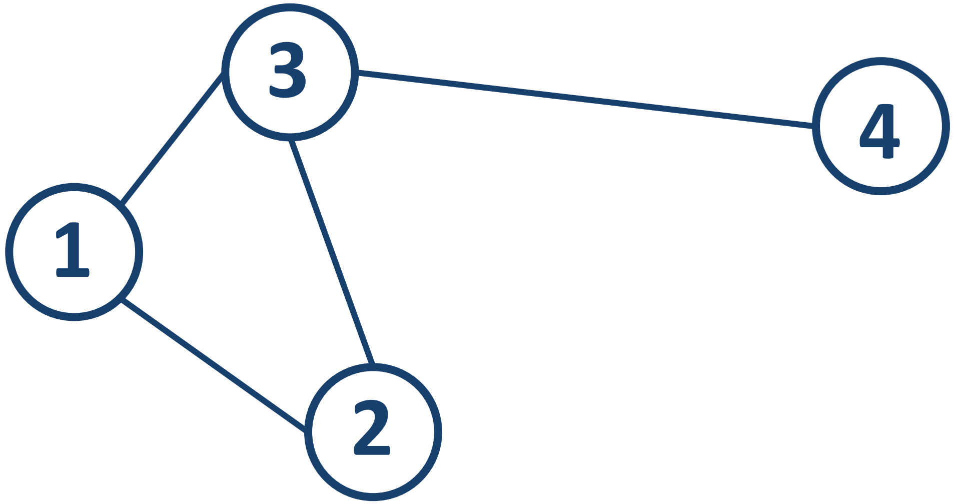


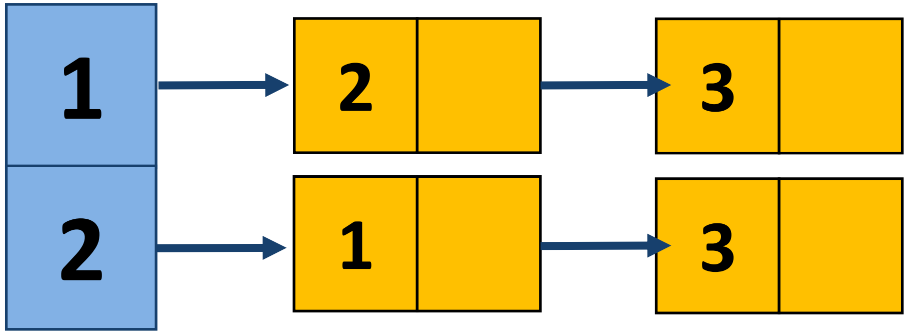
B-CPE-200 – Linked lists



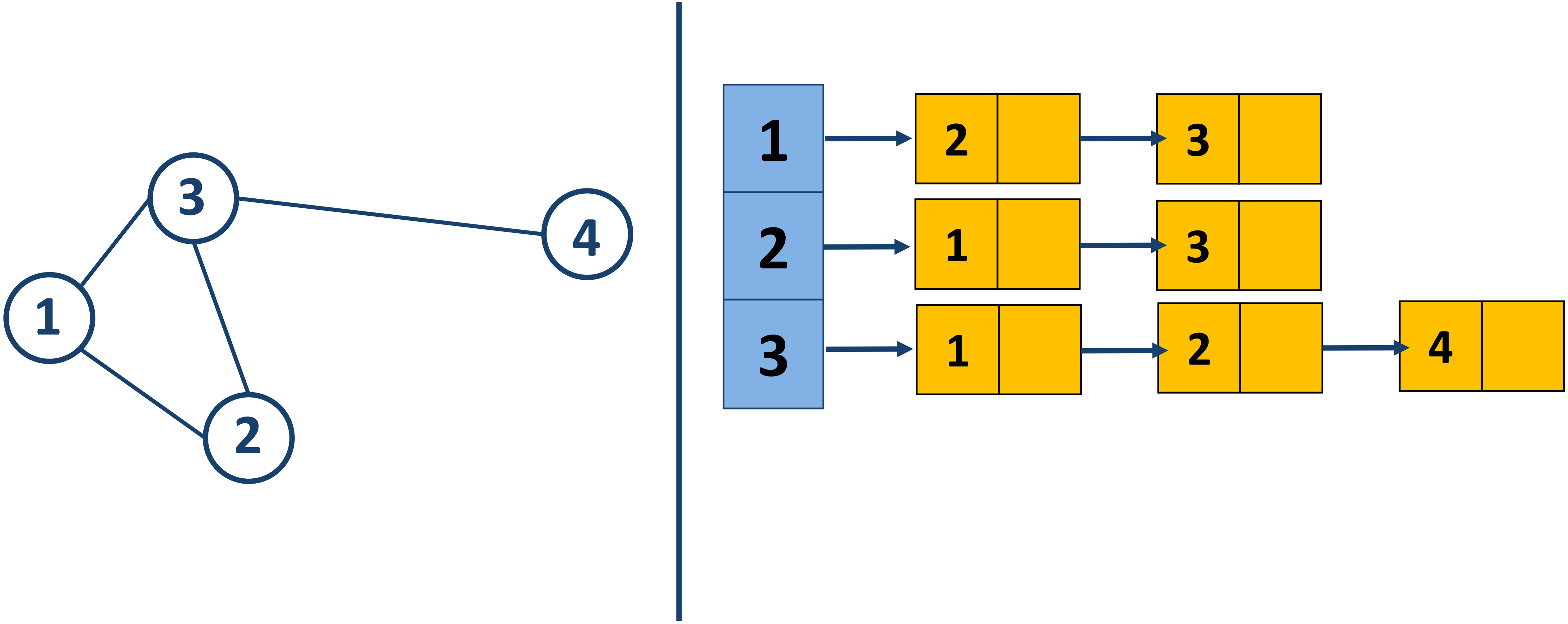


B-CPE-200 – Linked lists

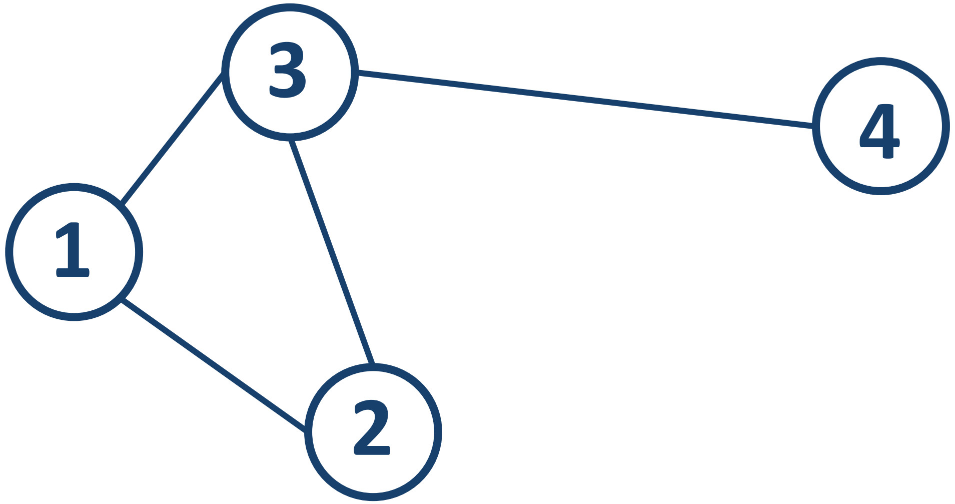


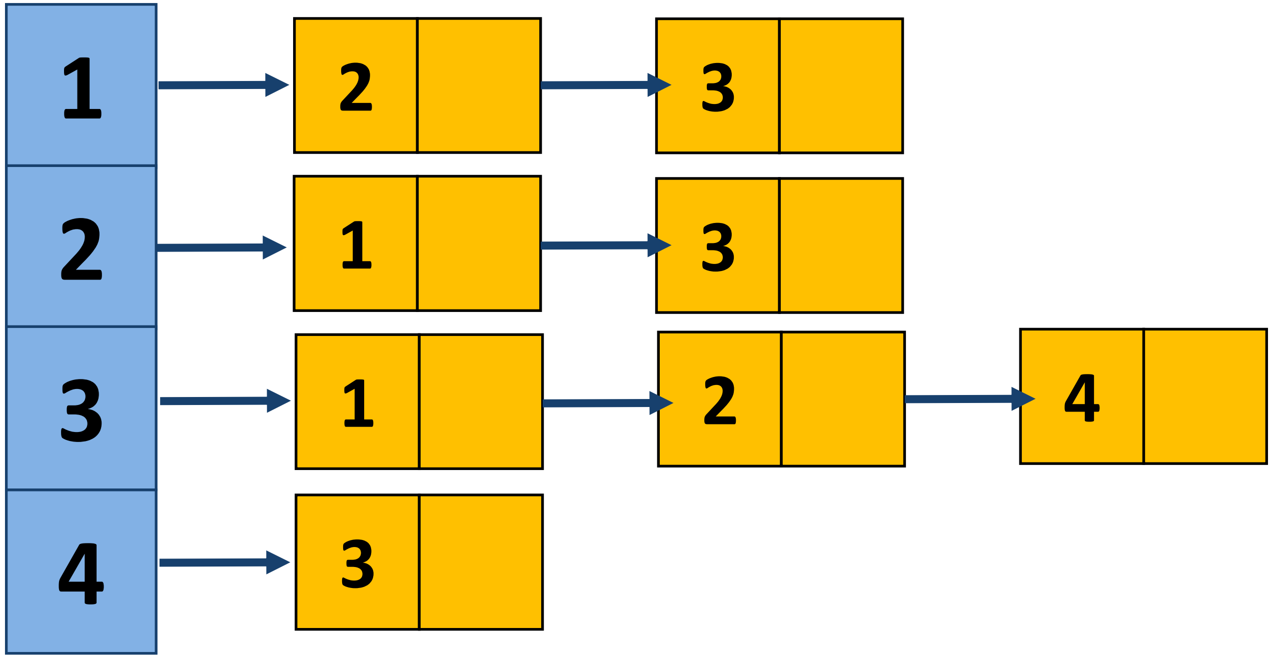


B-CPE-200 – Linked lists

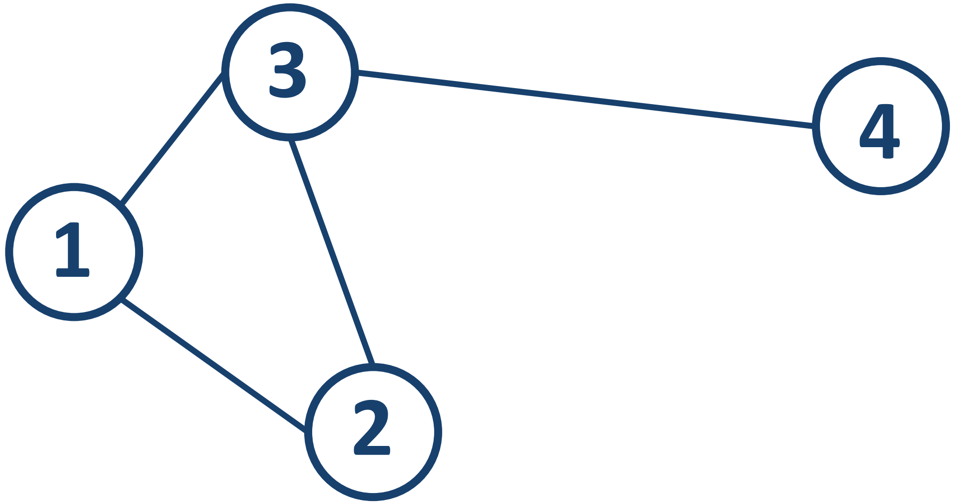


B-CPE-200 – Linked lists



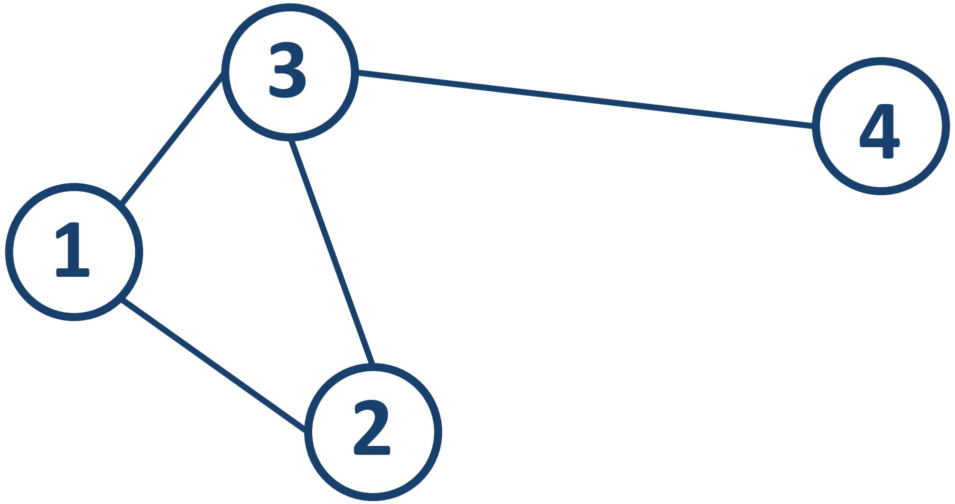


B-CPE-200 – Adjacency matrix



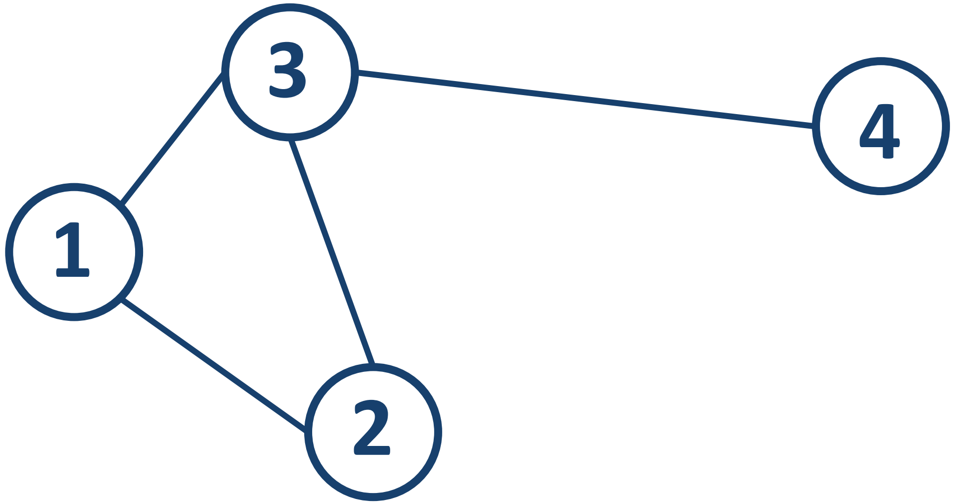
$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 \end{array}$$

B-CPE-200 – Adjacency matrix



$$\begin{array}{ccccc} & 1 & 2 & 3 & 4 \\ 1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 \end{array}$$

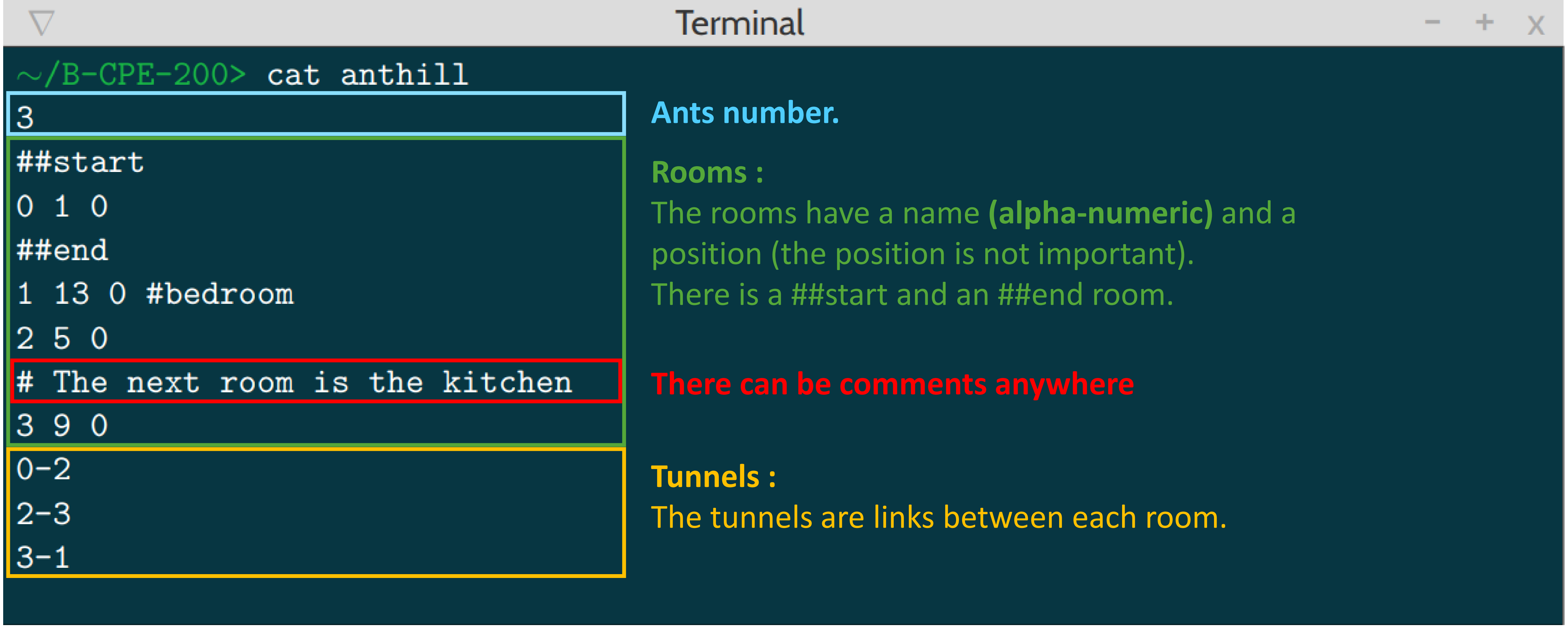
B-CPE-200 – Adjacency matrix



$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1 & 1 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ 3 & 1 & 1 & 0 & 1 \\ 4 & 0 & 0 & 1 & 0 \end{array}$$

Config

B-CPE-200 – Config



```
~/B-CPE-200> cat anthill
3
##start
0 1 0
##end
1 13 0 #bedroom
2 5 0
# The next room is the kitchen
3 9 0
0-2
2-3
3-1
```

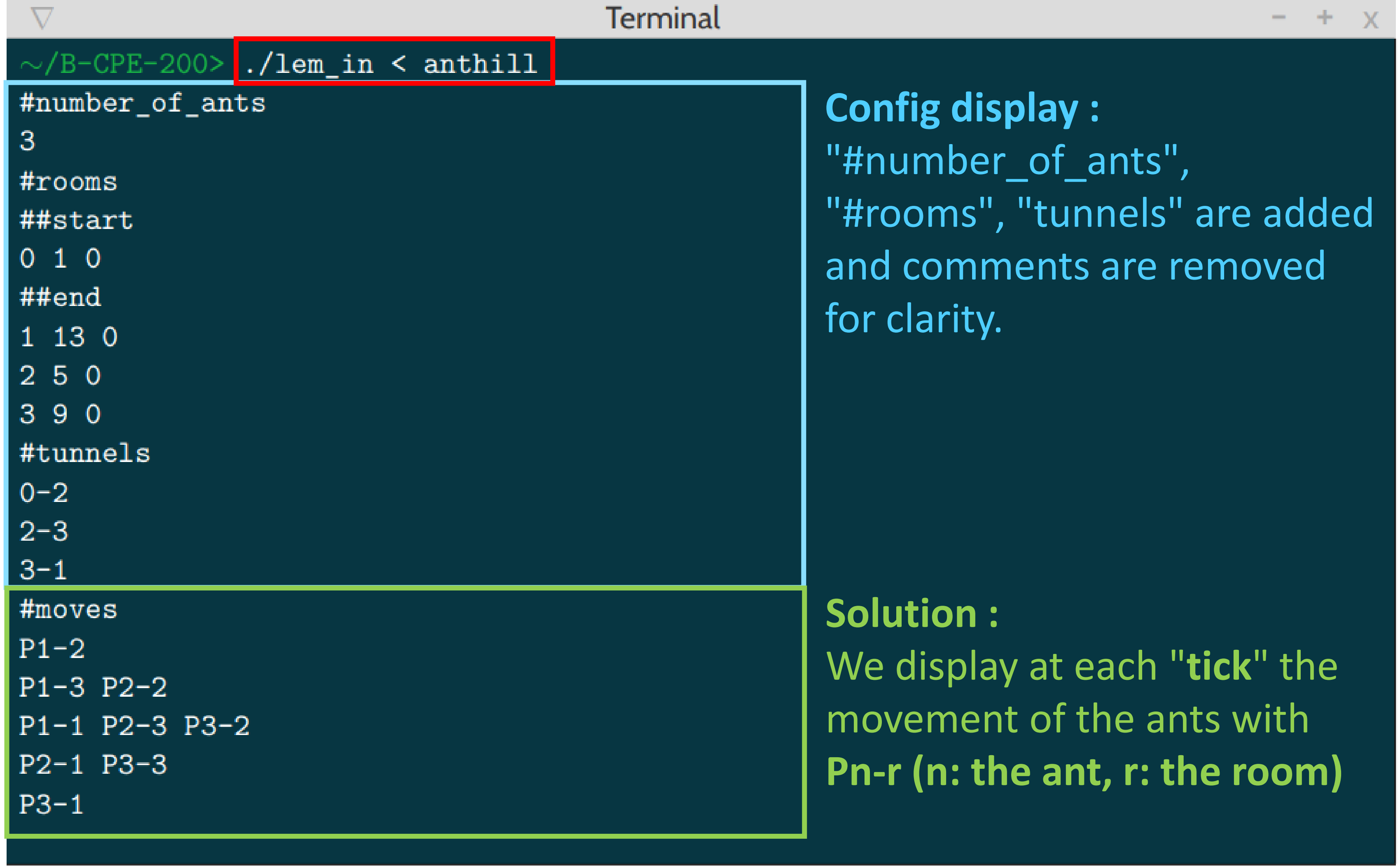
Ants number.

Rooms :
The rooms have a name **(alpha-numeric)** and a position (the position is not important).
There is a ##start and an ##end room.

There can be comments anywhere

Tunnels :
The tunnels are links between each room.

B-CPE-200 – Config



```
~/B-CPE-200> ./lem_in < anthill
#number_of_ants
3
#rooms
##start
0 1 0
##end
1 13 0
2 5 0
3 9 0
#tunnels
0-2
2-3
3-1
#moves
P1-2
P1-3 P2-2
P1-1 P2-3 P3-2
P2-1 P3-3
P3-1
```

Config display :
"#number_of_ants", "#rooms", "tunnels" are added and comments are removed for clarity.

Solution :
We display at each "**tick**" the movement of the ants with **Pn-r (n: the ant, r: the room)**

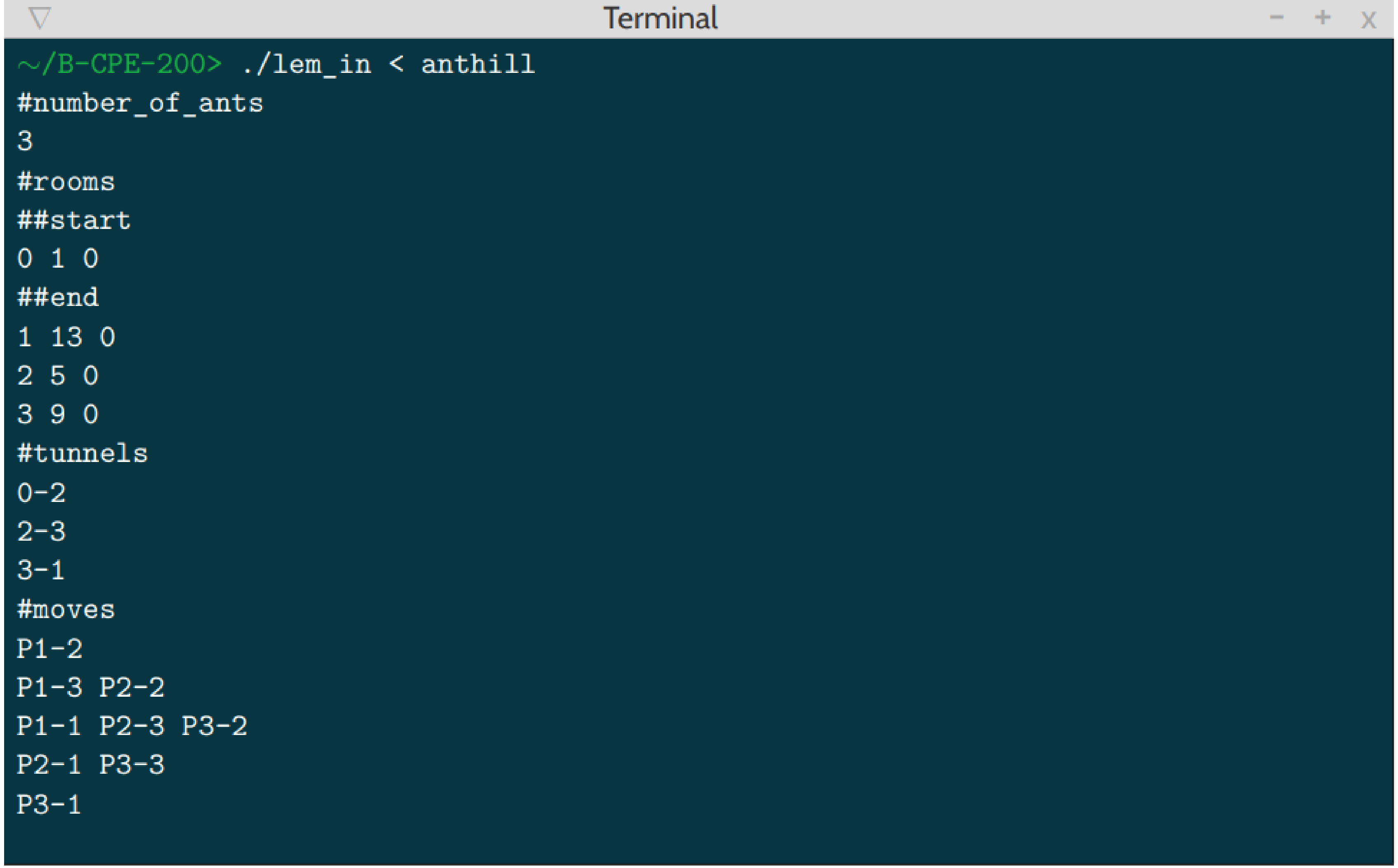
The **Parsing** skill is an important part of the project

Pathfinding is the most important skill of the project

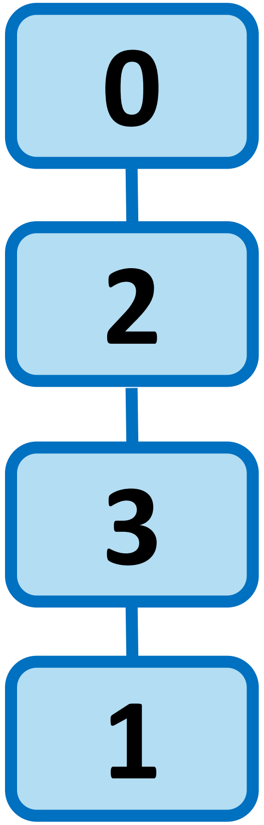
Be careful about the **Error Management**

A "tick" represents the ability of each ant to move one square

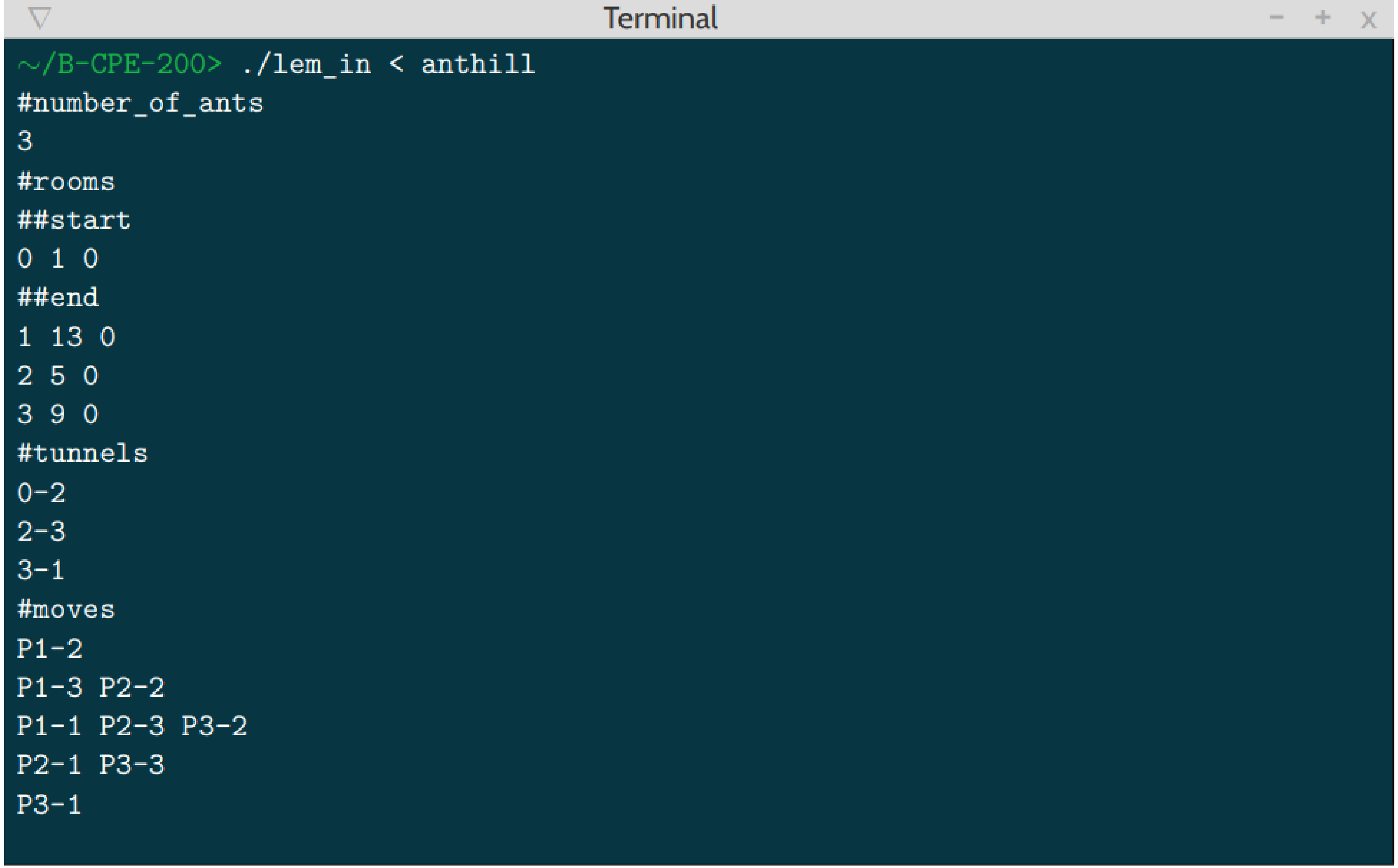
B-CPE-200 – Example



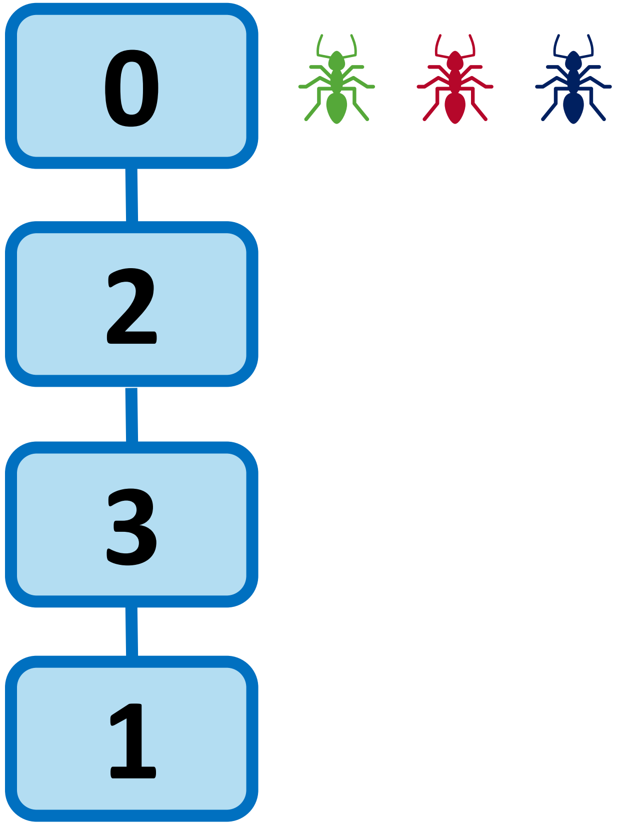
```
~/B-CPE-200> ./lem_in < anthill
#number_of_ants
3
#rooms
##start
0 1 0
##end
1 13 0
2 5 0
3 9 0
#tunnels
0-2
2-3
3-1
#moves
P1-2
P1-3 P2-2
P1-1 P2-3 P3-2
P2-1 P3-3
P3-1
```



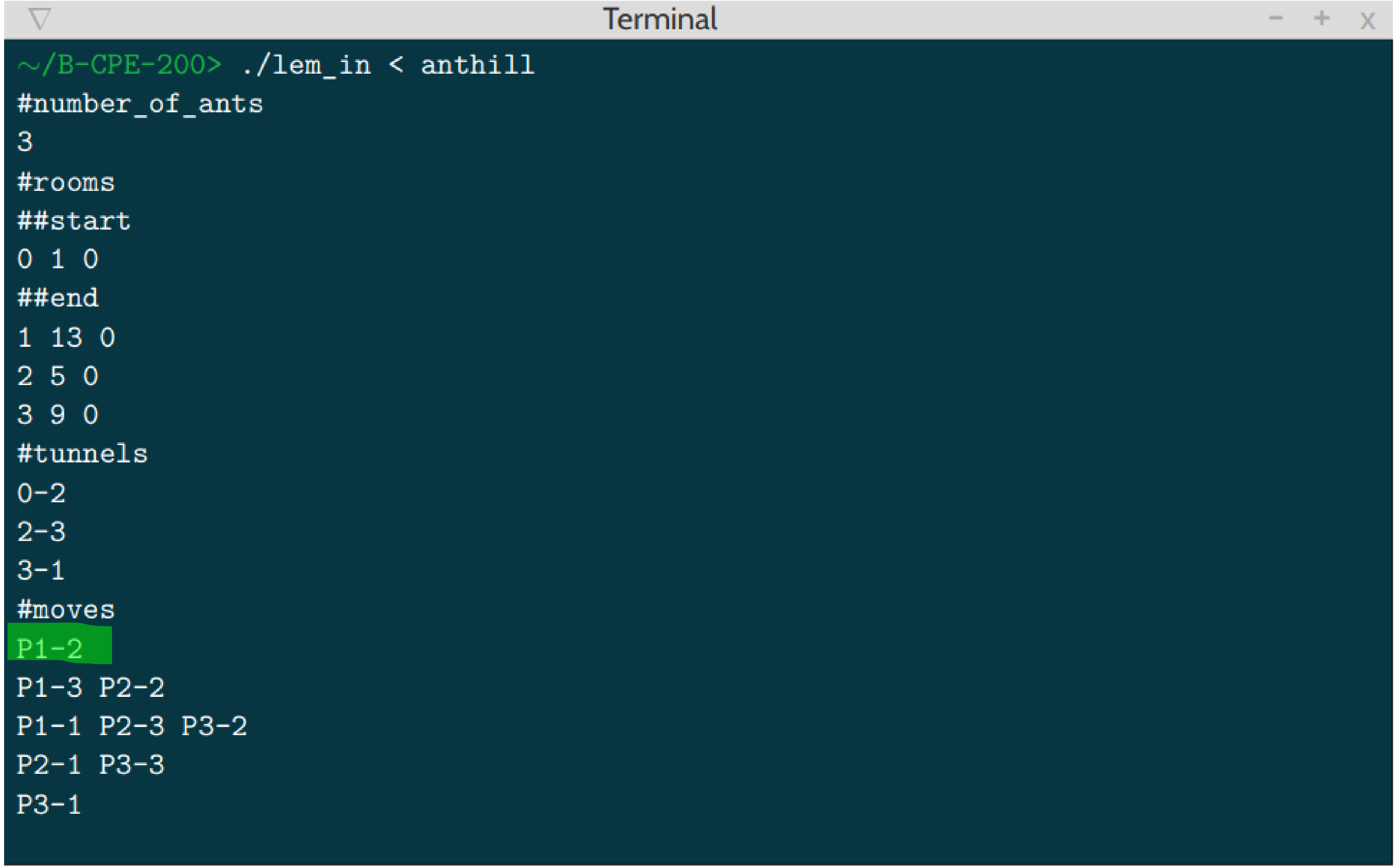
B-CPE-200 – Example



```
~/B-CPE-200> ./lem_in < anthill
#number_of_ants
3
#rooms
##start
0 1 0
##end
1 13 0
2 5 0
3 9 0
#tunnels
0-2
2-3
3-1
#moves
P1-2
P1-3 P2-2
P1-1 P2-3 P3-2
P2-1 P3-3
P3-1
```

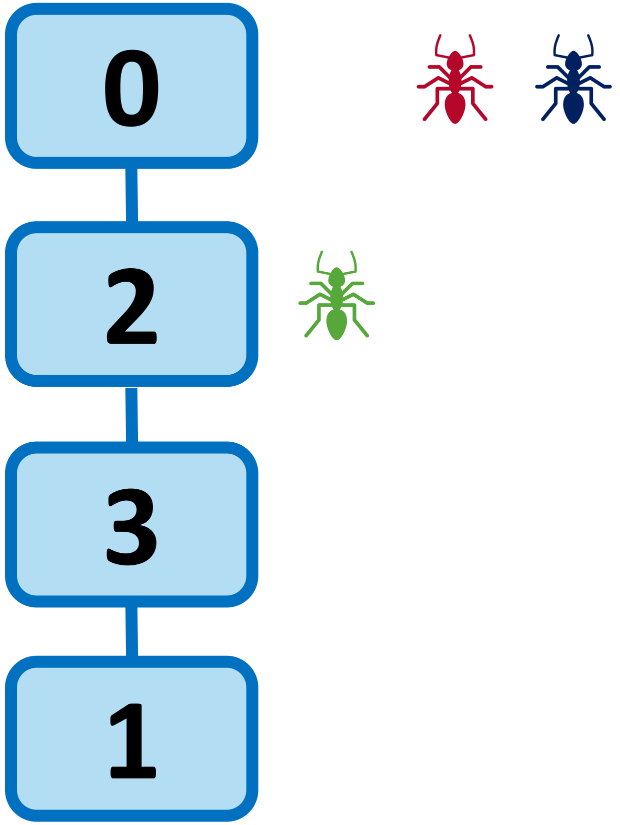


B-CPE-200 – Example

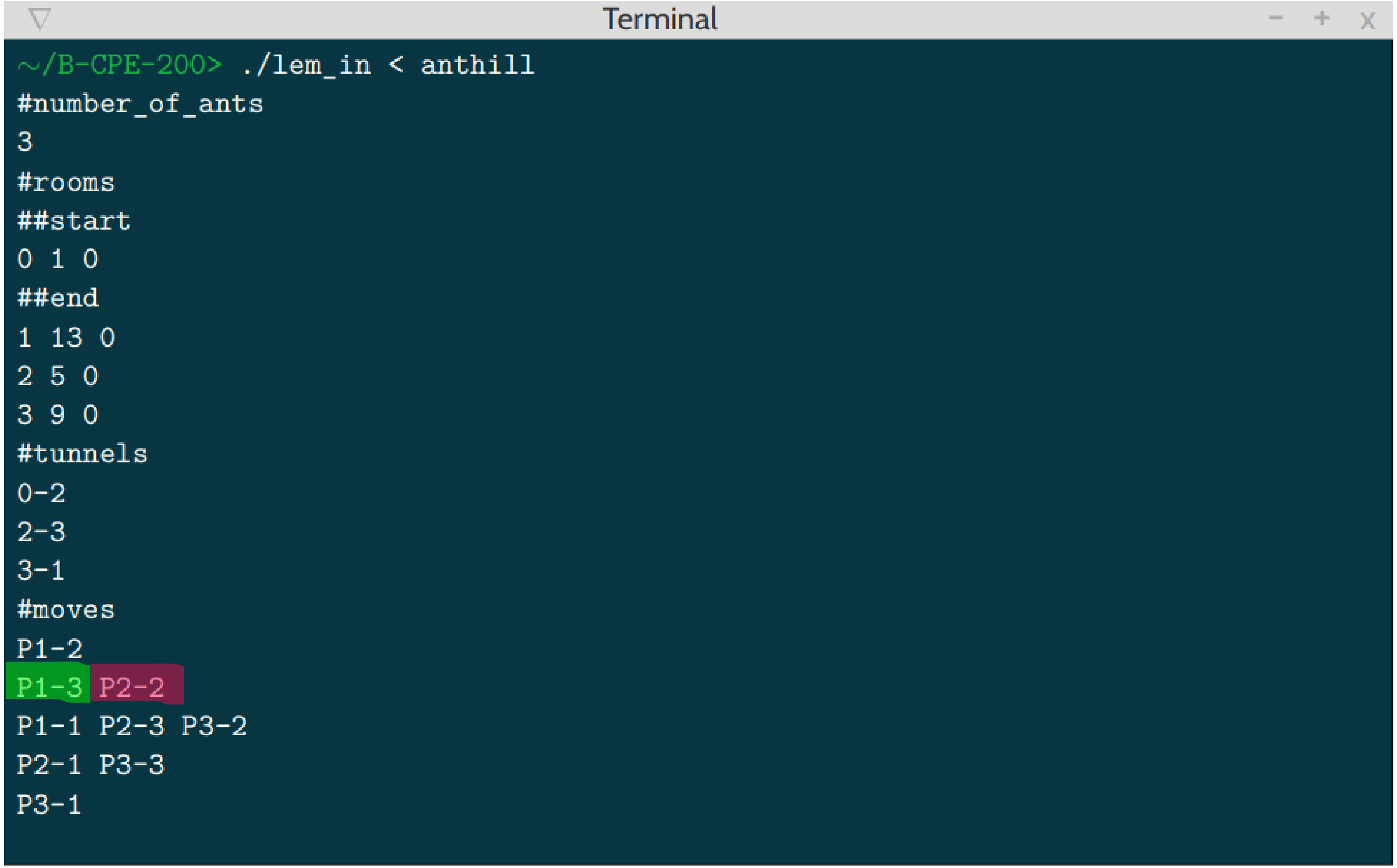


```
~/B-CPE-200> ./lem_in < anthill
#number_of_ants
3
#rooms
##start
0 1 0
##end
1 13 0
2 5 0
3 9 0
#tunnels
0-2
2-3
3-1
#moves
P1-2
P1-3 P2-2
P1-1 P2-3 P3-2
P2-1 P3-3
P3-1
```

P1-2

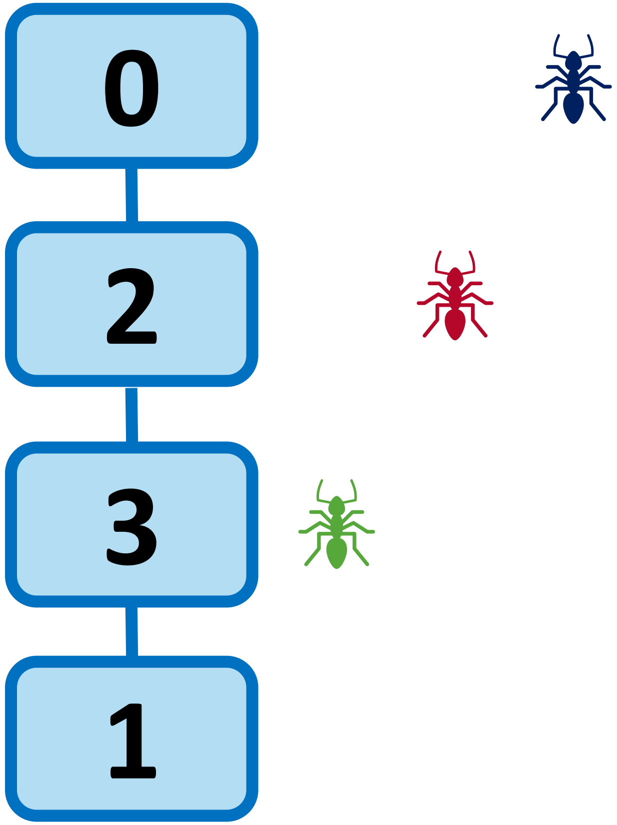


B-CPE-200 – Example

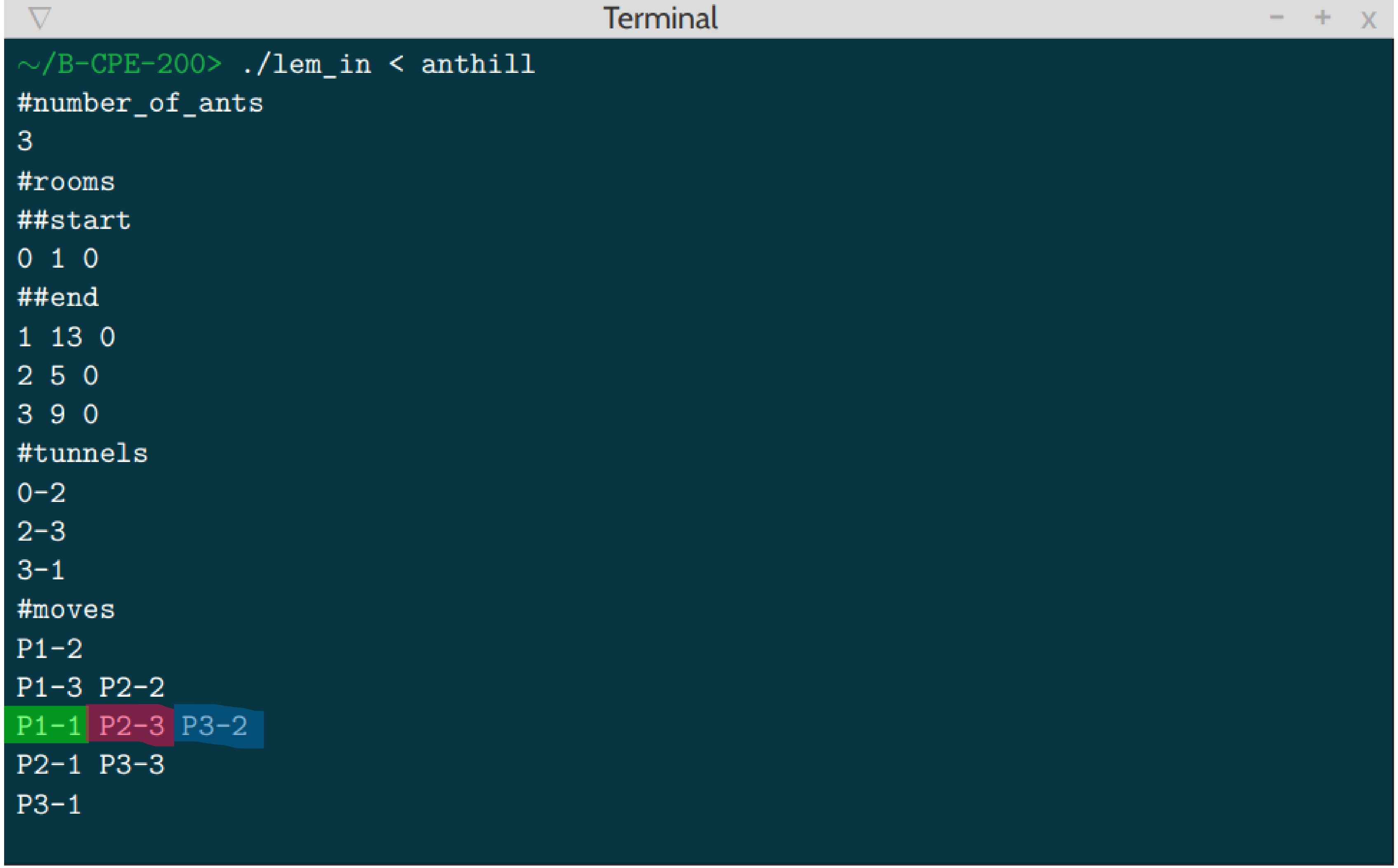


```
~/B-CPE-200> ./lem_in < anthill
#number_of_ants
3
#rooms
##start
0 1 0
##end
1 13 0
2 5 0
3 9 0
#tunnels
0-2
2-3
3-1
#moves
P1-2
P1-3 P2-2
P1-1 P2-3 P3-2
P2-1 P3-3
P3-1
```

P1-3, P2-2

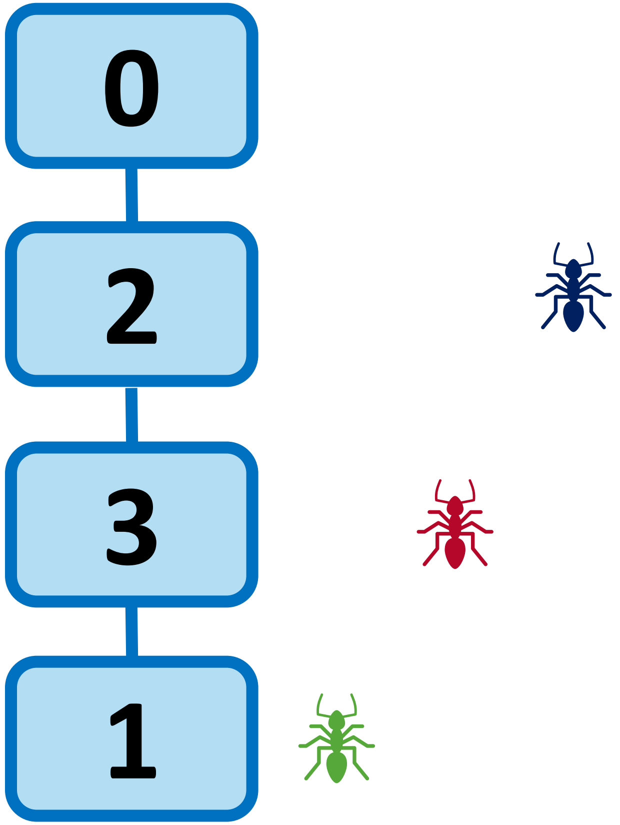


B-CPE-200 – Example

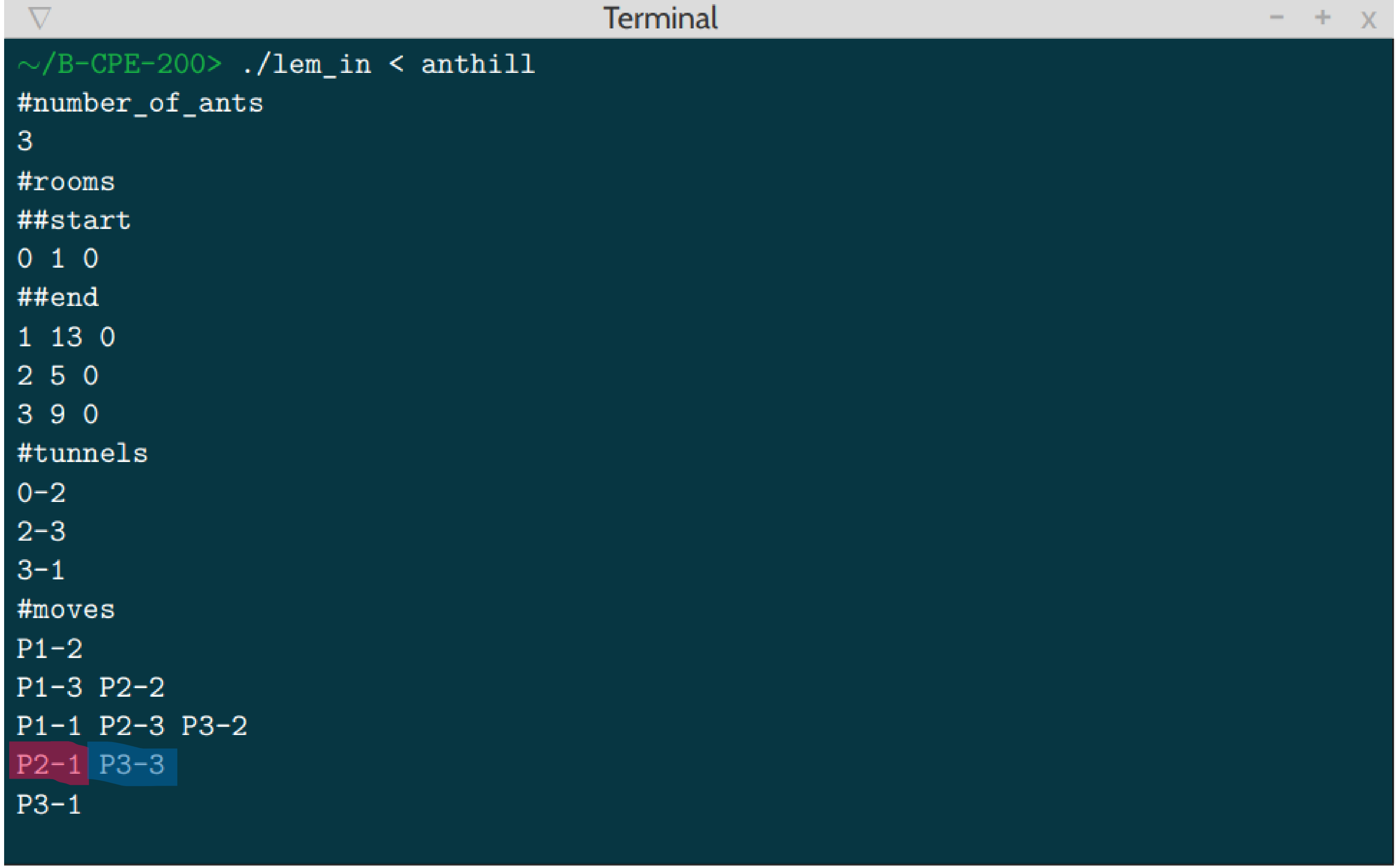


```
~/B-CPE-200> ./lem_in < anthill
#number_of_ants
3
#rooms
##start
0 1 0
##end
1 13 0
2 5 0
3 9 0
#tunnels
0-2
2-3
3-1
#moves
P1-2
P1-3 P2-2
P1-1 P2-3 P3-2
P2-1 P3-3
P3-1
```

P1-1, P2-3, P3-2

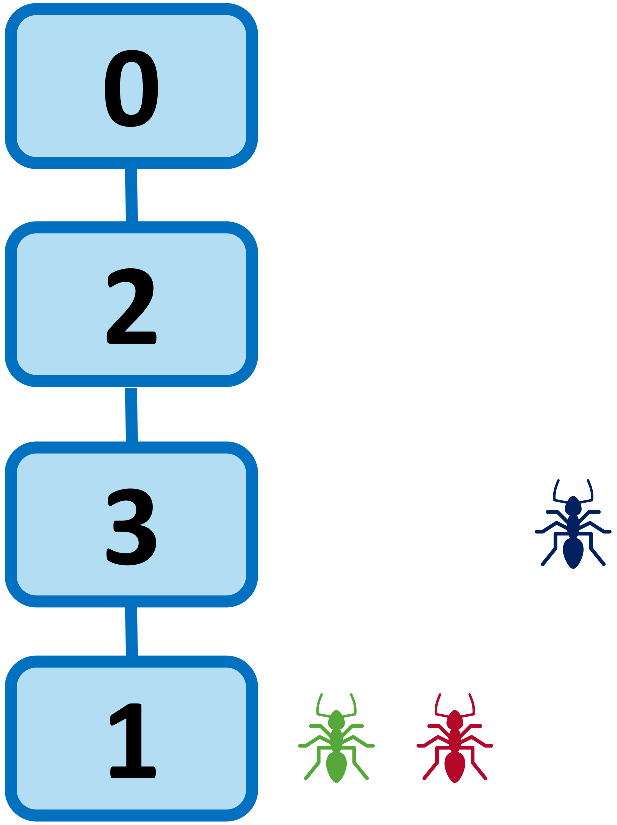


B-CPE-200 – Example

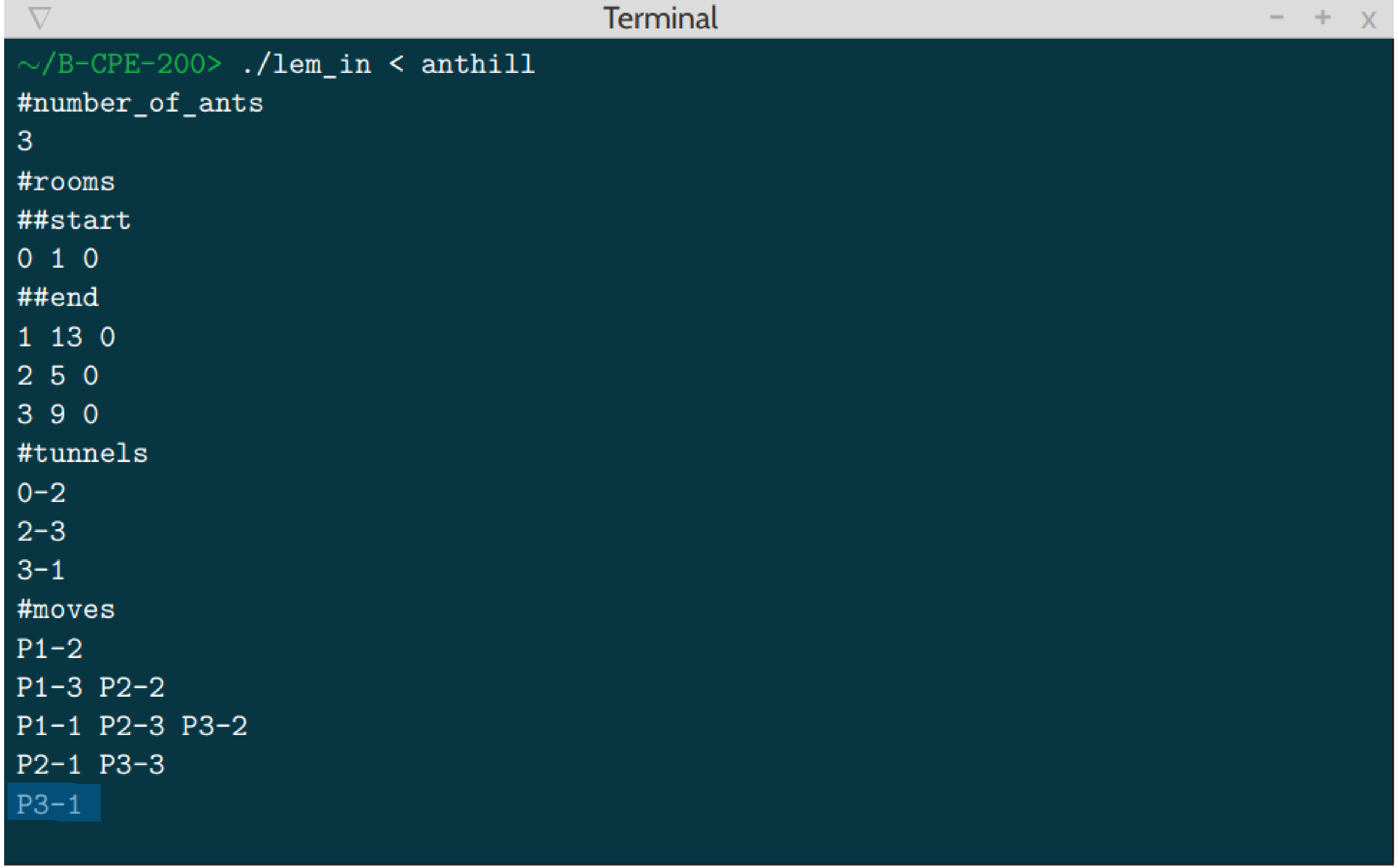


```
~/B-CPE-200> ./lem_in < anthill
#number_of_ants
3
#rooms
##start
0 1 0
##end
1 13 0
2 5 0
3 9 0
#tunnels
0-2
2-3
3-1
#moves
P1-2
P1-3 P2-2
P1-1 P2-3 P3-2
P2-1 P3-3
P3-1
```

P2-1, P3-3

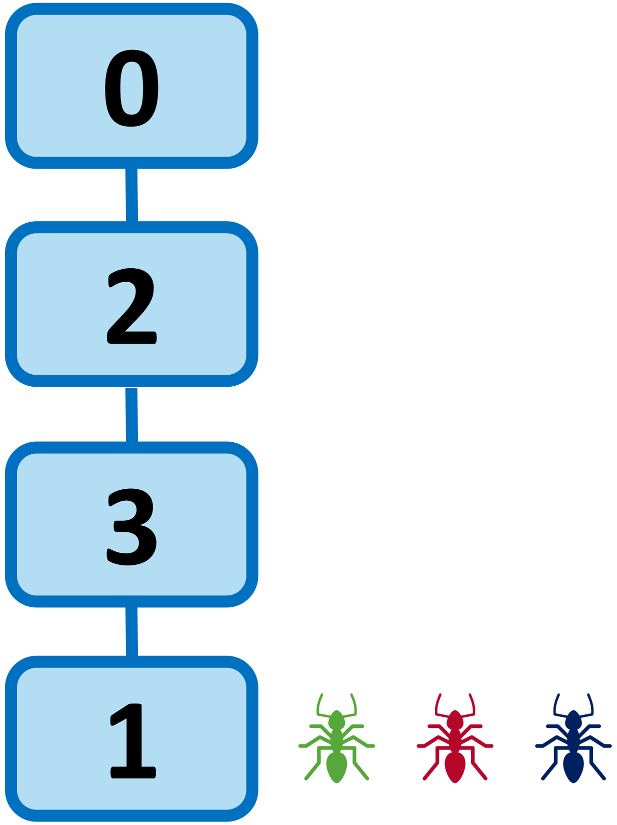


B-CPE-200 – Example



```
~/B-CPE-200> ./lem_in < anthill
#number_of_ants
3
#rooms
##start
0 1 0
##end
1 13 0
2 5 0
3 9 0
#tunnels
0-2
2-3
3-1
#moves
P1-2
P1-3 P2-2
P1-1 P2-3 P3-2
P2-1 P3-3
P3-1
```

P3-1



B-CPE-200 – Lem-in

What skills are evaluated?

Algorithm	**Data structure**	Parsing	Optimization	Robustness
Path finding algorithm	Graph application (adjacency matrix)	Yes	Ability to pass many ants	Error handling

{EPITECH. TECHNOLOGY}

Questions ?