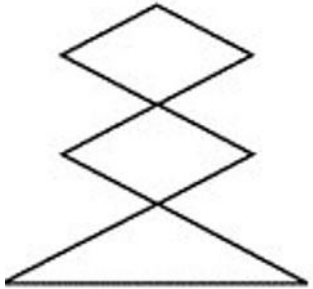
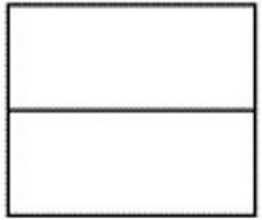
Power of Prolog & Theory Understanding

Seoul AI Meetup

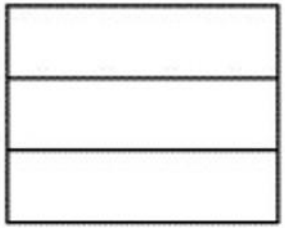
Martin Kersner, 2017/12/08



1)

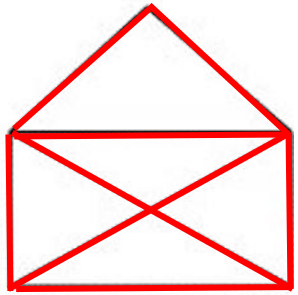


2)

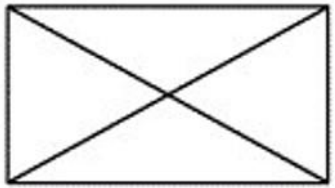


3)

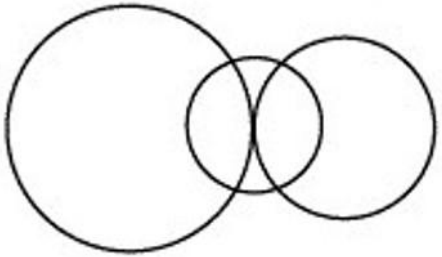
4)



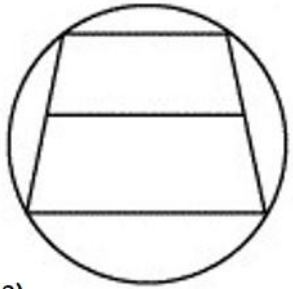
5)



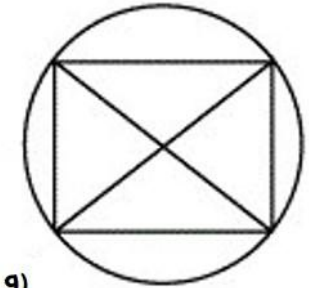
6)



7)

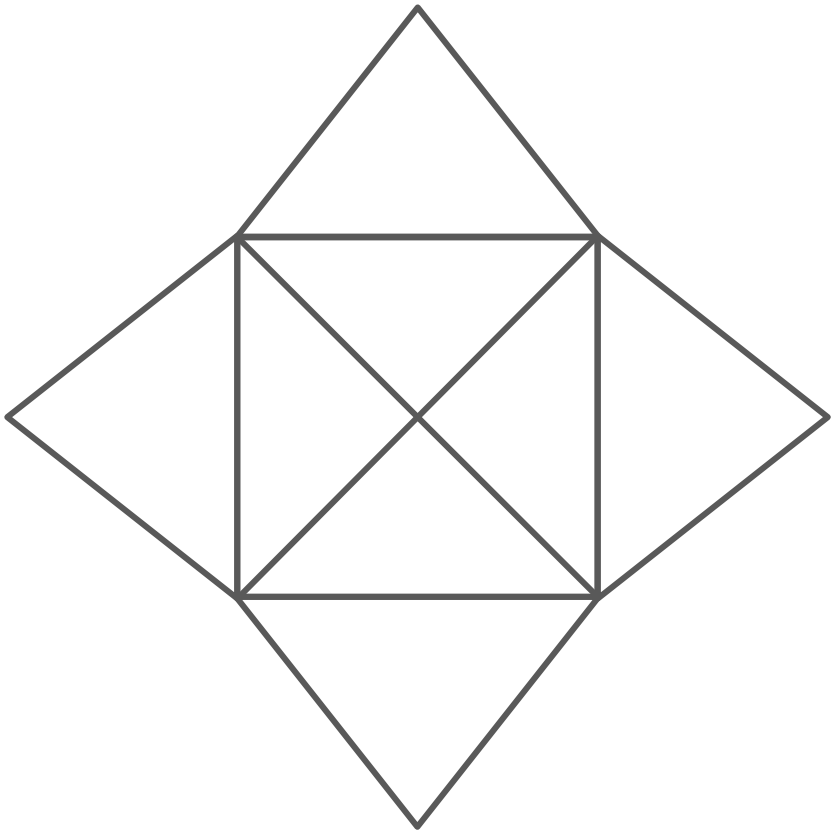


8)

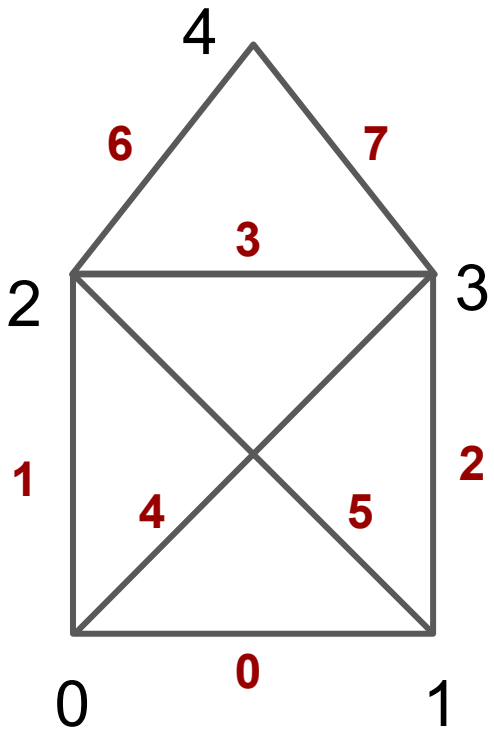


9)

“House” with 4 “roofs”



“House”



Prolog

Facts

Rules

Queries

“House” in Prolog representation

```
edge(0, 1, 0). % edge(node_id, node_id, edge_id)
edge(0, 2, 1).
edge(0, 3, 4).

edge(1, 2, 5).
edge(1, 3, 2).

edge(2, 3, 3).
edge(2, 4, 6).

edge(3, 4, 7).
```

Solution in Prolog

```
step([] , _).

step([H|T], C) :- (edge(N, C, H); (edge(C, N, H))),
                  step(T, N).

solve(E, [H|T]) :- permutation([H|T], E),
                   edge(C, _, H),
                   step([H|T], C).
```

C current node; **N** next node; **H** head of list; **T** tail of list; **E** all edges

Find all solutions for “House” problem

```
?- solve([0,1,2,3,4,5,6,7], Solution).
Solution = [0, 5, 1, 4, 3, 6, 7, 2] ;
Solution = [0, 5, 1, 4, 7, 6, 3, 2] ;
Solution = [0, 2, 3, 1, 4, 7, 6, 5]
```

Prolog finds one solution at the time and then waits for input.

; (semicolon) ask for searching another solution

. (dot) stop searching

Find all solutions starting with 0,5 and 1 edges

```
?- solve([0,1,2,3,4,5,6,7], [0,5,1,_,_,_,_,_]).
[0,5,1,4,3,6,7,2]
true ;
[0,5,1,4,7,6,3,2]
true ;
false.
```

“House” with 4 “roofs”

```
edge(0, 1, 0).
edge(0, 2, 1).

edge(1, 2, 2).
edge(1, 3, 3).
edge(1, 5, 4).
edge(1, 6, 5).

edge(2, 4, 6).
edge(2, 5, 7).
edge(2, 6, 8).
...
```

```
...
edge(3, 5, 9).

edge(4, 6, 10).

edge(5, 7, 11).
edge(5, 6, 12).

edge(6, 7, 13).
```

Find all solutions for “house” with 4 “roofs” problem

```
?- solve([0,1,2,3,4,5,6,7,8,9,10,11,12,13], Solution).
^CAction (h for help) ? abort
% Execution Aborted
```

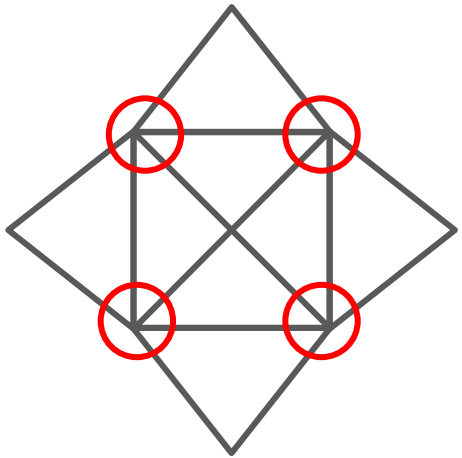
Execution takes **too long time**! Why?

8! = 40,320

14! = 87,178,291,200

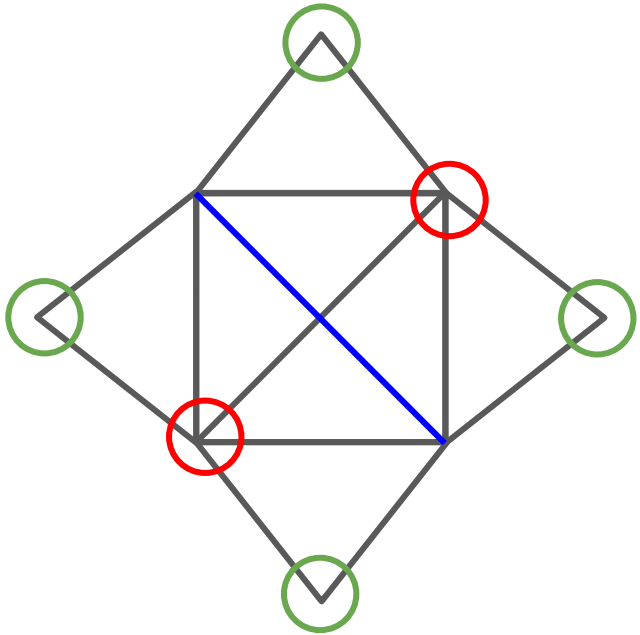
Help from Graph Theory, Eulerian Path

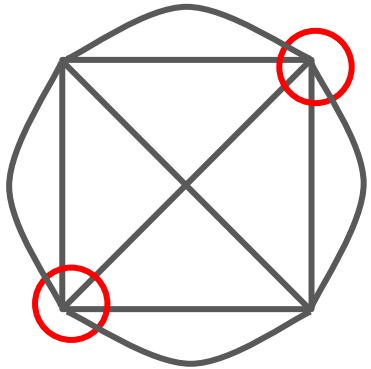
Euler proved that as long as a graph has either 0 or 2* vertices of odd degree, and the graph is connected (consists of a single piece), then it can be traversed through without visiting any of its edges twice.



4 odd degree vertices!

* If it has two 2 vertices of odd degree, then it must start at one, and end at the other.
https://puzzling.stackexchange.com/questions/12599/drawing-something-using-one-pen-stroke





(14-1)! = 6,227,020,800 >> (14-4)! = 3,628,800

References

https://www.matchilling.com/introduction-to-logic-programming-with-prolog/

https://www.metalevel.at/prolog

http://www.swi-prolog.org/

https://www.cs.nmsu.edu/ALP/wp-content/uploads/2011/03/PrologAndWatson1.pdf