IT5005 Artificial Intelligence
Tutorial 4

1. Rewrite the following clauses in CNF. Which clauses are Horn Clauses, which are Definite Clauses, and which are neither?

   a. $p \leftrightarrow q \vee r$

$$
\begin{aligned}
& p \leftrightarrow q \vee r \\
\equiv\ & (p \rightarrow q \vee r) \wedge (q \vee r \rightarrow p) \\
\equiv\ & (\sim \mathrm{p} \vee \mathrm{q} \vee \mathrm{r}) \wedge (\sim(\mathrm{q} \vee \mathrm{r}) \vee \mathrm{p}) \\
\equiv\ & (\sim p \vee q \vee r) \wedge \big((\sim q \wedge \sim r) \vee p\big) \\
\equiv\ & (\sim p \vee q \vee r) \wedge \big((\sim q \vee p) \wedge (\sim r \vee p)\big) \\
\equiv\ & (\sim p \vee q \vee r) \wedge (p \vee \sim q) \wedge (p \vee \sim r)
\end{aligned}
$$

   b. $q \vee r \rightarrow s$

$$
\begin{aligned}
& q \vee r \rightarrow s \\
\equiv\ & \sim(q \vee r) \vee s \\
\equiv\ & (\sim q \wedge \sim r) \vee s \\
\equiv\ & (\sim q \vee s) \wedge (\sim r \vee s)
\end{aligned}
$$

   c. $p \wedge q \rightarrow p \vee q$

$$
\begin{aligned}
& p \wedge q \rightarrow p \vee q \\
\equiv\ & \sim(p \wedge q) \vee (p \vee q) \\
\equiv\ & (\sim p \vee \sim q) \vee (p \vee q) \\
\equiv\ & \sim p \vee \sim q \vee p \vee q \\
\equiv\ & \sim p \vee p \vee \sim q \vee q \\
\equiv\ & True
\end{aligned}
$$

2. Recall the "Spectacle Problem" from lecture 2, reproduced here for your convenience:

   a. If I was reading the newspaper in the kitchen, then my glasses are on the kitchen table.
   b. If my glasses are on the kitchen table, then I saw them at breakfast.
   c. I did not see my glasses at breakfast.
   d. I was reading the newspaper in the living room or I was reading the newspaper in the kitchen.
   e. If I was reading the newspaper in the living room then my glasses are on the coffee table.

   Let

   - *RK* = I was reading the newspaper in the kitchen.
   - *GK* = My glasses are on the kitchen table.
   - *SB* = I saw my glasses at breakfast.
   - *RL* = I was reading the newspaper in the living room.
   - *GC* = My glasses are on the coffee table.

i. Write out the statements in propositional logic.

$$RK \rightarrow GK$$
$$GK \rightarrow SB$$
$$\sim SB$$
$$RL \vee RK$$
$$RL \rightarrow GC$$

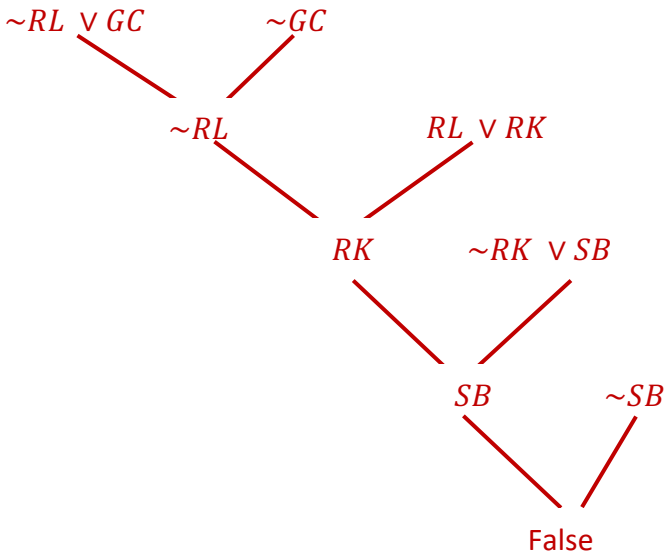
ii. Convert the statements into CNF form.

$$\sim RK \vee GK$$
$$\sim RK \vee SB$$
$$\sim SB$$
$$RL \vee RK$$
$$\sim RL \vee GC$$

iii. Use resolution to prove that the glasses are on the coffee table.

Resolution

Hypothesis: ~GC



3. In the mythical country of `Fiveohohfive Land, a person can hold a taxi license if the person has at least five years of driving experience, has obtained a certificate in public transport safety and possesses a certificate of merit. A person can possess a certificate of safety if the person passes the certification exam. A person can obtain a certificate of merit if a person is accident free and has not committed a traffic offence for the past five years.

A person who has a merit certificate has been driving for at least 5 years.

Use the following predicates to answer the questions below:

CanHasLicense(x1): x1 can to hold a taxi operator license.
FiveYears(x2): x2 has five years of driving experience.
Certificate(x3): x3 has a certificate in public transport safety.
Merit(x4): x4 has a certificate of merit.
PassedExam(x5): x5 passed the certificate exam.
AccidentFree(x6): x6 has been accident free for the past 5 years
OffenceFree(x7): x7 has been traffic offence free for the past 5 years.

a. Write out the full set of rules as stated above using the predicates given.

$$FiveYears(x1) \wedge Certificate(x1) \wedge Merit(x1) \rightarrow CanHasLicense(x1)$$
$$PassedExam(x1) \rightarrow Certificate(x1)$$
$$AccidentFree(x1) \wedge OffenceFree(x1) \rightarrow Merit(x1)$$
$$Merit(x1) \rightarrow FiveYears(x1)$$

b. Suppose we know:

PassedExam(Bob)
AccidentFree(Bob)
OffenceFree(Bob)

Use unification and forward-chaining to prove that Bob is qualified to hold a taxi operator license. Show all the unification and inference steps.

$$PassedExam(x1) \rightarrow Certificate(x1)$$
$$PassedExam(Bob)$$
$$\{x1 \backslash Bob\}$$
$$PassedExam(Bob) \rightarrow Certificate(Bob)$$

$$Certificate(Bob)$$

$$AccidentFree(x1) \wedge OffenceFree(x1) \rightarrow Merit(x1)$$
$$AccidentFree(Bob)$$
$$OffenceFree(Bob)$$
$$\{x1 \backslash Bob\}$$
$$AccidentFree(Bob) \wedge OffenceFree(Bob) \rightarrow Merit(Bob)$$

$$Merit(Bob)$$

$$Merit(x1) \rightarrow FiveYears(x1)$$
$$\{x1 \backslash BoB\}$$
$$Merit(Bob) \rightarrow FiveYears(Bob)$$

$$FiveYears(Bob)$$

$$FiveYears(x1) \wedge Certificate(x1) \wedge Merit(x1) \rightarrow CanHasLicense(x1)$$
$$FiveYears(Bob)$$
$$Certificate(Bob)$$
$$Merit(Bob)$$
$$\{x1 \backslash Bob\}$$
$$FiveYears(Bob) \wedge Certificate(Bob) \wedge Merit(x) \rightarrow CanHasLicense(Bob)$$

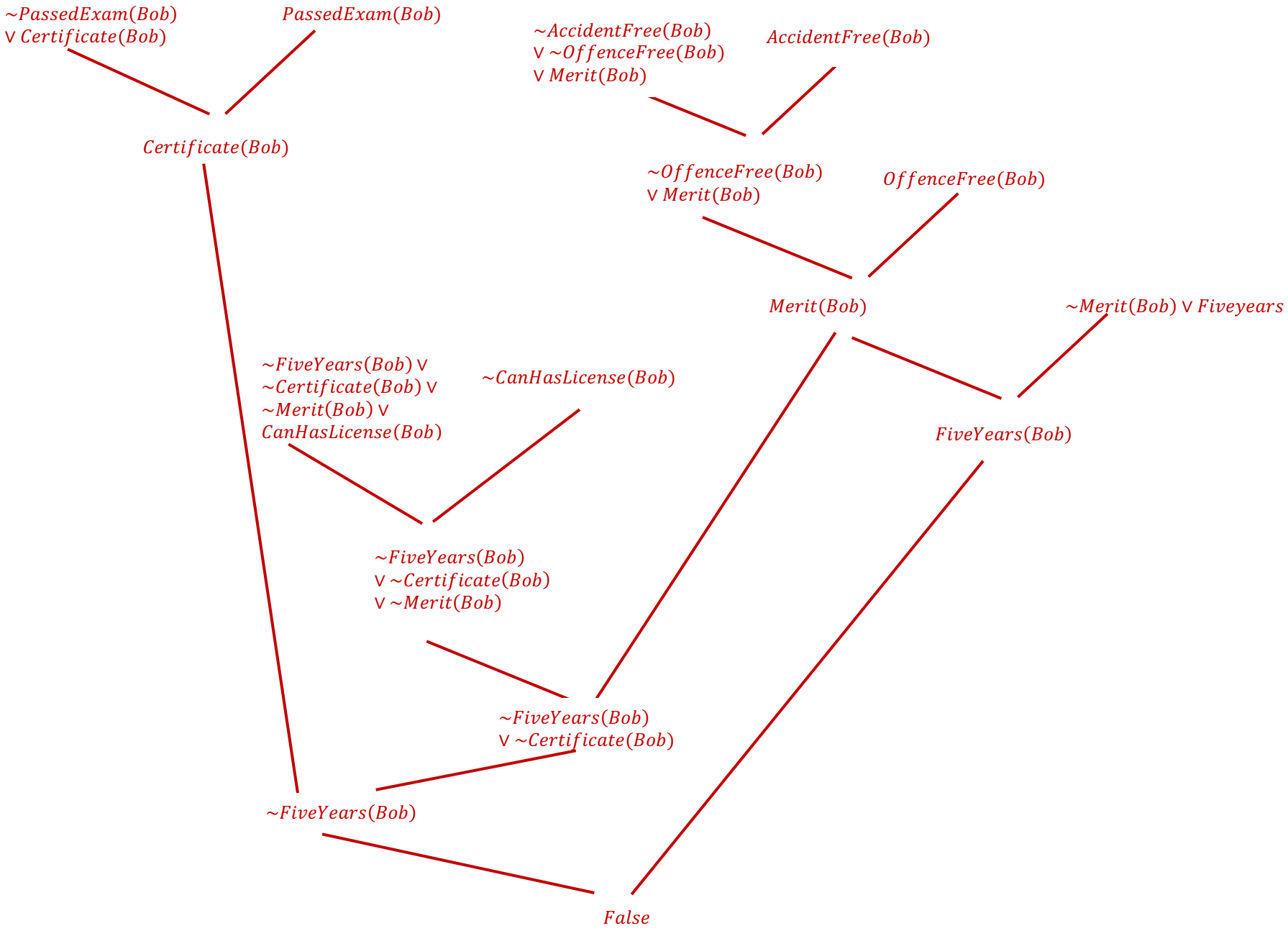
$$CanHasLicense(Bob)$$

c. Rewrite the rules in CNF, and use unification and resolution to prove that Bob is qualified to hold a taxi operator license.

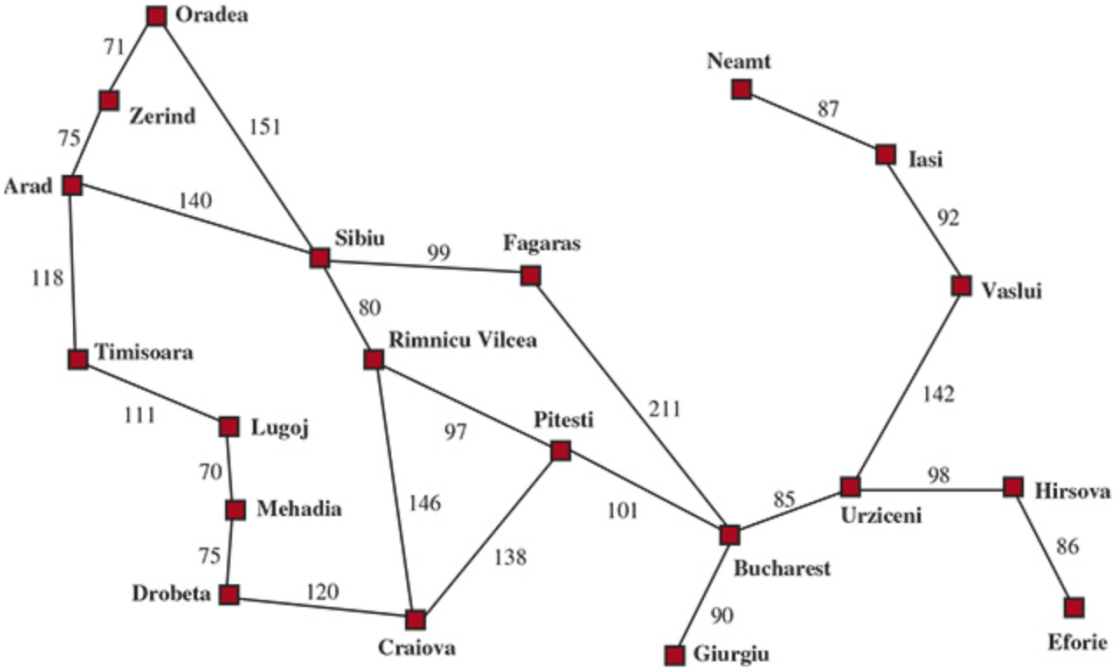
$$\sim FiveYears(x1) \vee \sim Certificate(x1) \vee \sim Merit(x1) \vee CanHasLicense(x1)$$
$$\sim PassedExam(x1) \vee Certificate(x1)$$
$$\sim AccidentFee(x1) \vee \sim OffenceFree(x1) \vee Merit(x1)$$
$$\sim Merit(x1) \vee FiveYears(x1)$$

$$\sim CanHasLicense(Bob)$$
$$\{x1 \backslash Bob\}$$
$$\sim FiveYears(Bob) \vee \sim Certificate(Bob) \vee \sim Merit(Bob) \vee CanHasLicense(Bob)$$

From here on we assume that all variables have been unified with Bob.

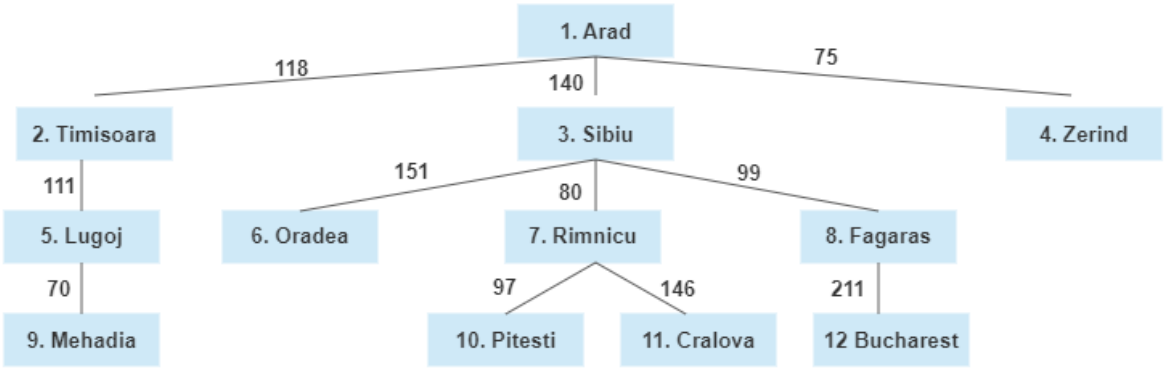

~PassedExam(Bob) ∨ Certificate(Bob)
PassedExam(Bob)
Certificate(Bob)
~AccidentFree(Bob) ∨ ~OffenceFree(Bob) ∨ Merit(Bob)
AccidentFree(Bob)
~OffenceFree(Bob) ∨ Merit(Bob)
OffenceFree(Bob)
Merit(Bob)
~Merit(Bob) ∨ Fiveyears
~FiveYears(Bob) ∨ ~Certificate(Bob) ∨ ~Merit(Bob) ∨ CanHasLicense(Bob)
~CanHasLicense(Bob)
FiveYears(Bob)
~FiveYears(Bob) ∨ ~Certificate(Bob) ∨ ~Merit(Bob)
~FiveYears(Bob) ∨ ~Certificate(Bob)
~FiveYears(Bob)
False

4. We come back to our Romanian Problem, where we want to get from Arad to Bucharest using the map shown below:



Apply Depth First Search, Breadth First Search and Dijkstra's Algorithm to produce a route from Arad to Bucharest, showing the distance of the route for each algorithm.

Breadth First Search



Route: Arad->Sibiu->Fagaras->Bucharest. Total distance = 140 + 99 + 211 = 450 miles

Depth First Search

(Omitted. Please try on your own)

Dijkstra's Algorithm

Step	V(T)	E(T)	F	L(Arad)	L(Zerind)	L(Timisoara)	L(Sibiu)	L(Lugoj)	L(Oradea)	L(Fagaras)	L(Mehadia)	L(Drobeta)	L(Rimicu)	L(Pitesti)	L(Craolva)	L(Bucharest)
0	{Arad}	Empty	{Arad}	0	inf	inf	inf	inf	inf	inf	Inf	Inf	inf	inf	Imf	inf
1	{Arad}	Empty	{Zerind, Sibiu, Timisoara)	0	75	118	140	inf	inf	inf	Inf	Inf	inf	inf	Inf	inf
2	{Arad, Zerind}	{(Arad, Zerind)	{Sibiu, Timisoara, Oradea)	0	75	118	140	inf	146	inf	Inf	Inf	inf	inf	Inf	inf
3	{Arad, Zerind, Timisoara}	{(Arad, Zerind), (Arad, Timisoara)	{Sibiu, Oradea, Lugoj)	0	75	118	140	229	146	inf	Inf	Inf	inf	inf	Inf	inf
4	{Arad, Zerind, Timisoara, Sibiu}	{(Arad, Zerind), (Arad, Timisoara), (Arad, Sibiu)}	{Oradea, Lugoj, Fagaras, Rimicu)	0	75	118	140	229	146	239	Inf	Inf	220	inf	Inf	inf
5	{Arad, Zerind, Timisoara, Sibu, Oradea}	{(Arad, Zerind), (Arad, Timisoara), (Arad, Sibiu), (Zerind, Oradea)}	{Lugoj, Fagaras, Rimicu}	0	75	118	140	229	146	239	Inf	Inf	220	Inf	Inf	Inf
6	{Arad, Zerind, Timisoara, Sibu, Oradea, Rimicu}	{(Arad, Zerind), (Arad, Timisoara), (Arad, Sibiu), (Zerind, Oradea), (Sibiu, Rimicu)}	{Lugoj, Fagaras, Pitesti, Cralova}	0	75	118	140	229	146	239	Inf	Inf	220	317	366	Inf

Step	V(T)	E(T)	F	L(Arad)	L(Zerind)	L(Timisoara)	L(Sibiu)	L(Lugoj)	L(Oradea)	L(Fagaras)	L(Mehadia)	L(Drobeta)	L(Rimicu)	L(Pitesti)	L(Craolva)	L(Bucharest)
7	{Arad, Zerind, Timisoara, Sibu, Oradea, Rimicu, Lugoj}	{(Arad, Zerind), (Arad, Timisoara), (Arad, Sibiu), (Zerind, Oradea), (Sibiu, Rimicu), (Timisoara, Lugoj)}	{Fagaras, Pitesti, Cralova, Mehadia}	0	75	118	140	229	146	239	299	Inf	220	317	366	Inf
8	Arad, Zerind, Timisoara, Sibu, Oradea, Rimicu, Lugoj, Fagaras}	{(Arad, Zerind), (Arad, Timisoara), (Arad, Sibiu), (Zerind, Oradea), (Sibiu, Rimicu), (Timisoara, Lugoj), (Sibiu, Fagaras)}	{Pitesti, Cralova, Mehadia Bucharest}	0	75	118	140	229	146	239	299	Inf	220	317	366	450
9	{Arad, Zerind, Timisoara, Sibu, Oradea, Rimicu, Lugoj, Fagaras, Mehadia}	{(Arad, Zerind), (Arad, Timisoara), (Arad, Sibiu), (Zerind, Oradea), (Sibiu, Rimicu), (Timisoara, Lugoj), (Sibiu, Fagaras), (Lugoj, Mehadia)}	{Pitesti, Cralova, Buchare. Drobetat}	0	75	118	140	229	146	239	299	374	220	317	366	450
10	{Arad, Zerind, Timisoara, Sibu, Oradea, Rimicu, Lugoj, Fagaras, Mehadia, Pitesti{	{(Arad, Zerind), (Arad, Timisoara), (Arad, Sibiu), (Zerind, Oradea), (Sibiu, Rimicu), (Timisoara, Lugoj), (Sibiu, Fagaras), (Lugoj, Mehadia)}	{ Cralova, Bucharest. Drobetat}	0	75	118	140	229	146	239	299	374	220	317	366	418
11	{Arad, Zerind, Timisoara, Sibu, Oradea, Rimicu, Lugoj, Fagaras, Mehadia, Pitesti}	{(Arad, Zerind), (Arad, Timisoara), (Arad, Sibiu), (Zerind, Oradea), (Sibiu, Rimicu), (Timisoara, Lugoj), (Sibiu, Fagaras), (Lugoj, Mehadia)}	{Cralova, Bucharest.}	0	75	118	140	229	146	239	299	374	220	317	366	418

Step	V(T)	E(T)	F	L(Arad)	L(Zerind)	L(Timisoara)	L(Sibiu)	L(Lugoj)	L(Oradea)	L(Fagaras)	L(Mehadia)	L(Drobeta)	L(Rimicu)	L(Pitesti)	L(Craolva)	L(Bucharest)
12	{Arad, Zerind, Timisoara, Sibu, Oradea, Rimicu, Lugoj, Fagaras, Mehadia, Pitesti}	{(Arad, Zerind), (Arad, Timisoara), (Arad, Sibiu), (Zerind, Oradea), (Sibiu, Rimicu), (Timisoara, Lugoj), (Sibiu, Fagaras), (Lugoj, Mehadia)}	{Bucharest.}	0	75	118	140	229	146	239	299	374	220	317	366	418
13	{Arad, Zerind, Timisoara, Sibu, Oradea, Rimicu, Lugoj, Fagaras, Mehadia, Pitesti, Bucharest}	{(Arad, Zerind), (Arad, Timisoara), (Arad, Sibiu), (Zerind, Oradea), (Sibiu, Rimicu), (Timisoara, Lugoj), (Sibiu, Fagaras), (Lugoj, Mehadia), (Pitesti, Bucharest)}	{Giurgu, Urziceni}	0	75	118	140	229	146	239	299	374	220	317	366	418

Distance: 418 miles