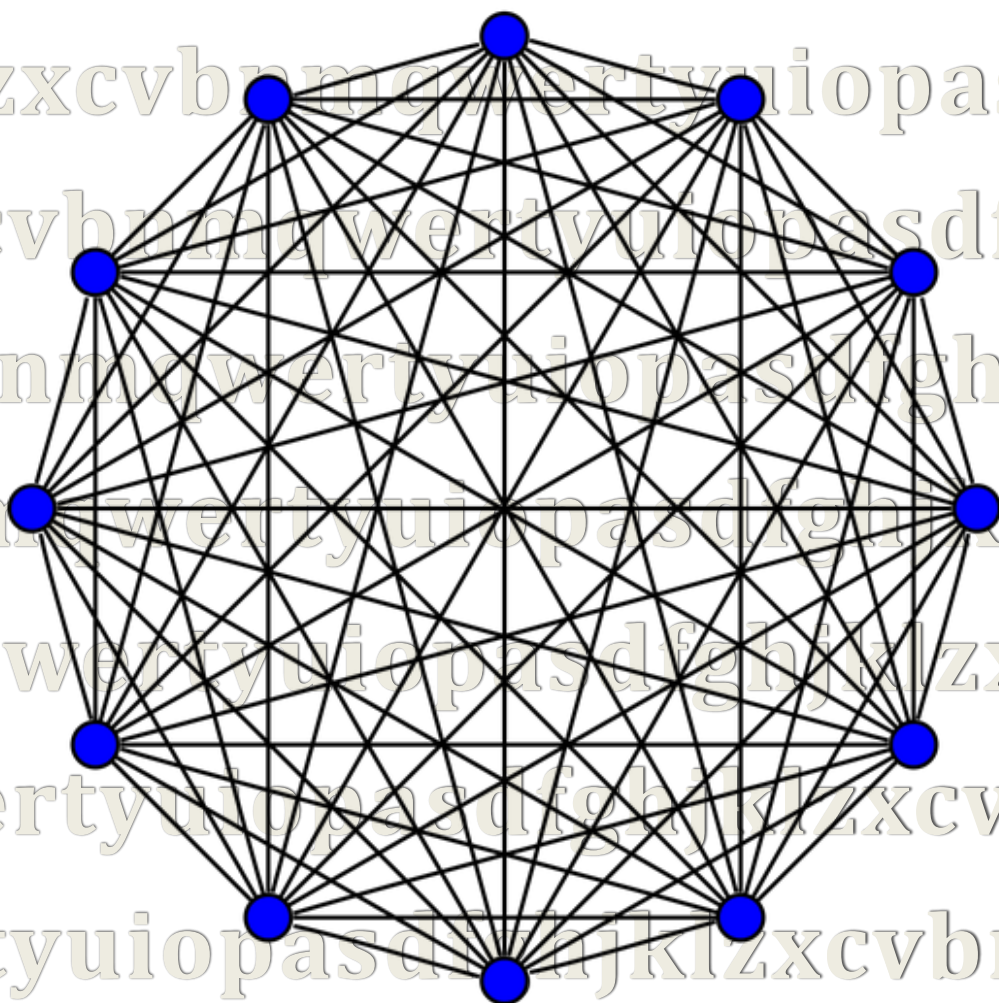


圖論作業

李文源、郭菱雅

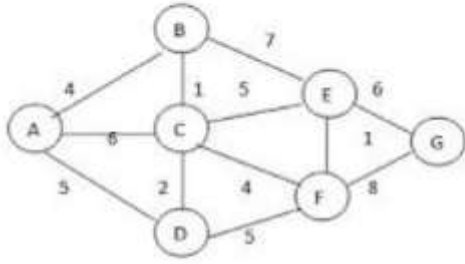
朱怡亭、黃珮綺

2014/6/5



圖論合作學習單

小組成員	黃珮綺、朱怡亭、郭菱雅、李文源
時間 / 地點	6/5. 7:00pm. S504
問題描述	● Kneser graph 問題 令 Kneser graph $KG(6,2)$ 代表以集合 $\{1,2,3,4,5,6\}$ 中之元素個數恰為 2 的所有子集合(一般稱之為"2-subset")當 vertices，任兩個相異的 2-subsets 如果 disjoint 則連一條 edge。 1. 試畫出 $KG(6,2)$ 並且在其上找出一條 Euler circuit。 2. 試畫出 $KG(6,2)$ 並且在其上找出一條 Hamilton cycle。 5. 請在 3×4 的西洋棋盤上找出一條能讓騎士恰好遍歷每一個格子一次且走回原來位置的路徑。如果可以的話，請找出所有的路徑。 ● Spanning Tree 問題 圖(一) 7. 請在圖(一)中利用 BFS algorithm 或 DFS algorithm 找出其 spanning tree。如果可以的話，請找出所有不同的 spanning tree。

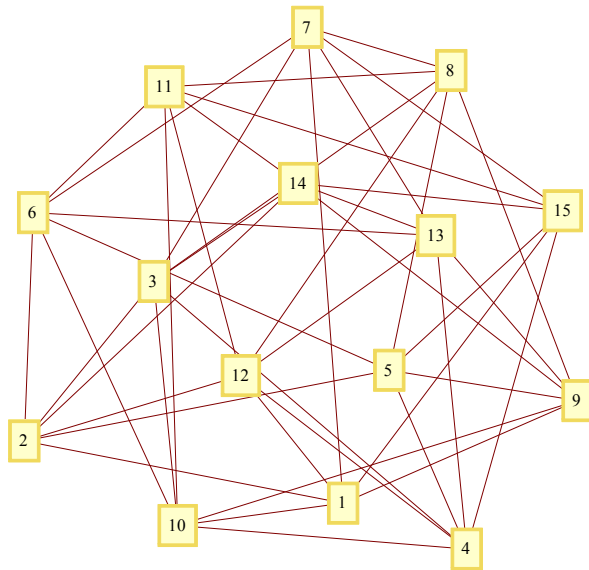
	● Minimum Spanning Tree 問題  圖(二) 9. 請在圖(二)中利用 Prim's algorithm 找出其 minimim spanning tree 。
討論過程	第 1.2 題 >> 黃珮綺 第 5 題 >> 李文源 第 7 題 >> 郭菱雅 第 9 題 >> 朱怡亭
結論	解題容易，但編寫程式不容易，有些題目有想法但卻無法順利完成程式，跑出結果。





KG (6, 2)

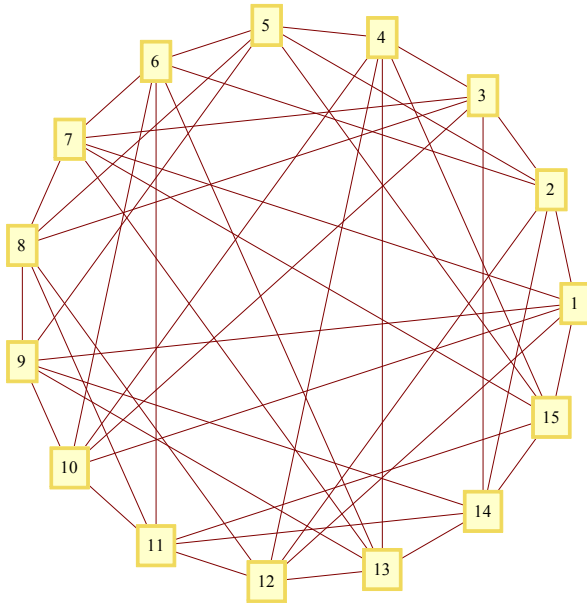
```
f = GraphPlot[{1 → 2, 2 → 3, 3 → 4, 4 → 5, 5 → 6, 6 → 7, 7 → 8, 8 → 9,  
  9 → 10, 10 → 11, 11 → 12, 12 → 13, 13 → 14, 14 → 15, 15 → 1, 1 → 7, 1 → 9,  
  1 → 10, 1 → 12, 2 → 5, 2 → 6, 2 → 12, 2 → 14, 3 → 7, 3 → 8, 3 → 10, 3 → 14,  
  4 → 10, 4 → 12, 4 → 13, 4 → 15, 5 → 8, 5 → 9, 5 → 15, 6 → 10, 6 → 11,  
  6 → 13, 7 → 13, 7 → 15, 8 → 11, 8 → 12, 9 → 13, 9 → 14, 11 → 14, 11 → 15},  
  SelfLoopStyle → 0.2, MultiedgeStyle → 0.2, VertexLabeling → True]
```



```

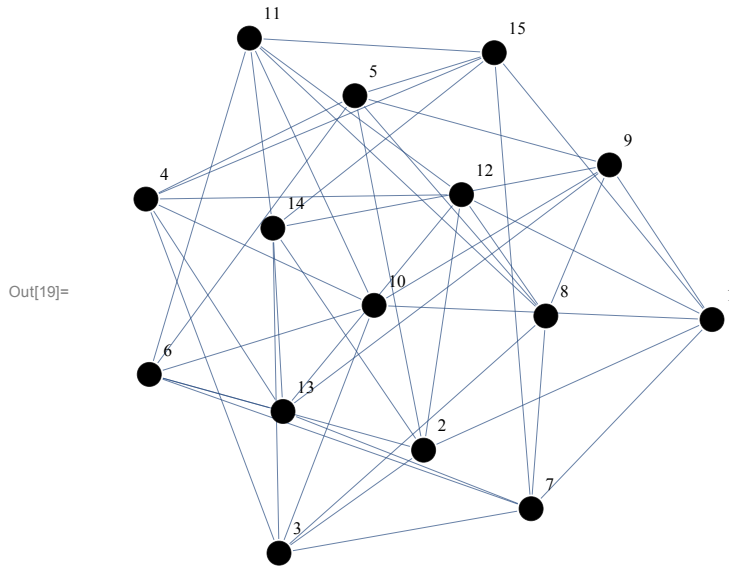
k = GraphPlot[ {1 → 2, 2 → 3, 3 → 4, 4 → 5, 5 → 6, 6 → 7, 7 → 8, 8 → 9, 9 → 10, 10 → 11,
11 → 12, 12 → 13, 13 → 14, 14 → 15, 15 → 1, 1 → 7, 1 → 9, 1 → 10, 1 → 12,
2 → 5, 2 → 6, 2 → 12, 2 → 14, 3 → 7, 3 → 8, 3 → 10, 3 → 14, 4 → 10, 4 → 12,
4 → 13, 4 → 15, 5 → 8, 5 → 9, 5 → 15, 6 → 10, 6 → 11, 6 → 13, 7 → 13, 7 → 15,
8 → 11, 8 → 12, 9 → 13, 9 → 14, 11 → 14, 11 → 15}, SelfLoopStyle → 0.2,
MultiedgeStyle → 0.2, VertexLabeling → True, VertexCoordinateRules →
{1 → {1, 0}, 2 → {Cos[24 °], Sin[24 °]}, 3 → {Sin[42 °], Cos[42 °]},
4 → { $\frac{1}{4}(-1 + \sqrt{5})$ ,  $\sqrt{\frac{5}{8} + \frac{\sqrt{5}}{8}}$ }, 5 → {-Sin[6 °], Cos[6 °]}, 6 → {- $\frac{1}{2}$ ,  $\frac{\sqrt{3}}{2}$ },
7 → { $\frac{1}{4}(-1 - \sqrt{5})$ ,  $\sqrt{\frac{5}{8} - \frac{\sqrt{5}}{8}}$ }, 8 → {-Cos[12 °], Sin[12 °]},
9 → {-Cos[12 °], -Sin[12 °]}, 10 → { $\frac{1}{4}(-1 - \sqrt{5})$ ,  $-\sqrt{\frac{5}{8} - \frac{\sqrt{5}}{8}}$ },
11 → {- $\frac{1}{2}$ ,  $-\frac{\sqrt{3}}{2}$ }, 12 → {-Sin[6 °], -Cos[6 °]}, 13 → { $\frac{1}{4}(-1 + \sqrt{5})$ ,  $-\sqrt{\frac{5}{8} + \frac{\sqrt{5}}{8}}$ },
14 → {Sin[42 °], -Cos[42 °]}, 15 → {Cos[24 °], -Sin[24 °]}}]

```



1. Euler circuit

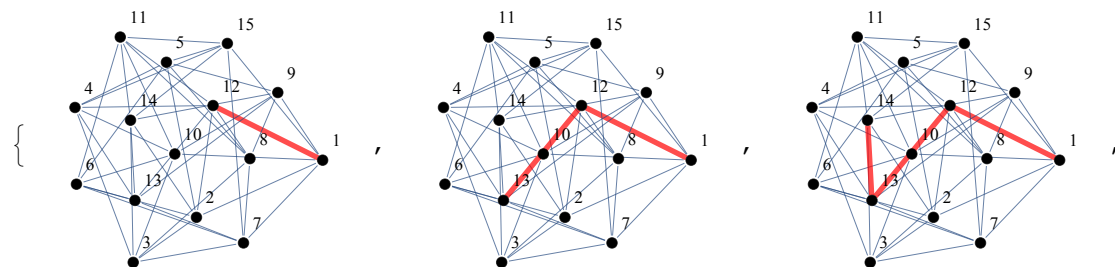
```
j = Graph[Range[7], {1 ↔ 2, 2 ↔ 3, 3 ↔ 4, 4 ↔ 5, 5 ↔ 6, 6 ↔ 7, 7 ↔ 8, 8 ↔ 9, 9 ↔ 10,
  10 ↔ 11, 11 ↔ 12, 12 ↔ 13, 13 ↔ 14, 14 ↔ 15, 15 ↔ 1, 1 ↔ 7, 1 ↔ 9, 1 ↔ 10, 1 ↔ 12,
  2 ↔ 5, 2 ↔ 6, 2 ↔ 12, 2 ↔ 14, 3 ↔ 7, 3 ↔ 8, 3 ↔ 10, 3 ↔ 14, 4 ↔ 10, 4 ↔ 12,
  4 ↔ 13, 4 ↔ 15, 5 ↔ 8, 5 ↔ 9, 5 ↔ 15, 6 ↔ 10, 6 ↔ 11, 6 ↔ 13, 7 ↔ 13, 7 ↔ 15,
  8 ↔ 11, 8 ↔ 12, 9 ↔ 13, 9 ↔ 14, 11 ↔ 14, 11 ↔ 15}, VertexLabels → "Name",
  VertexSize → Medium, VertexStyle → Black, ImagePadding → 20]
```

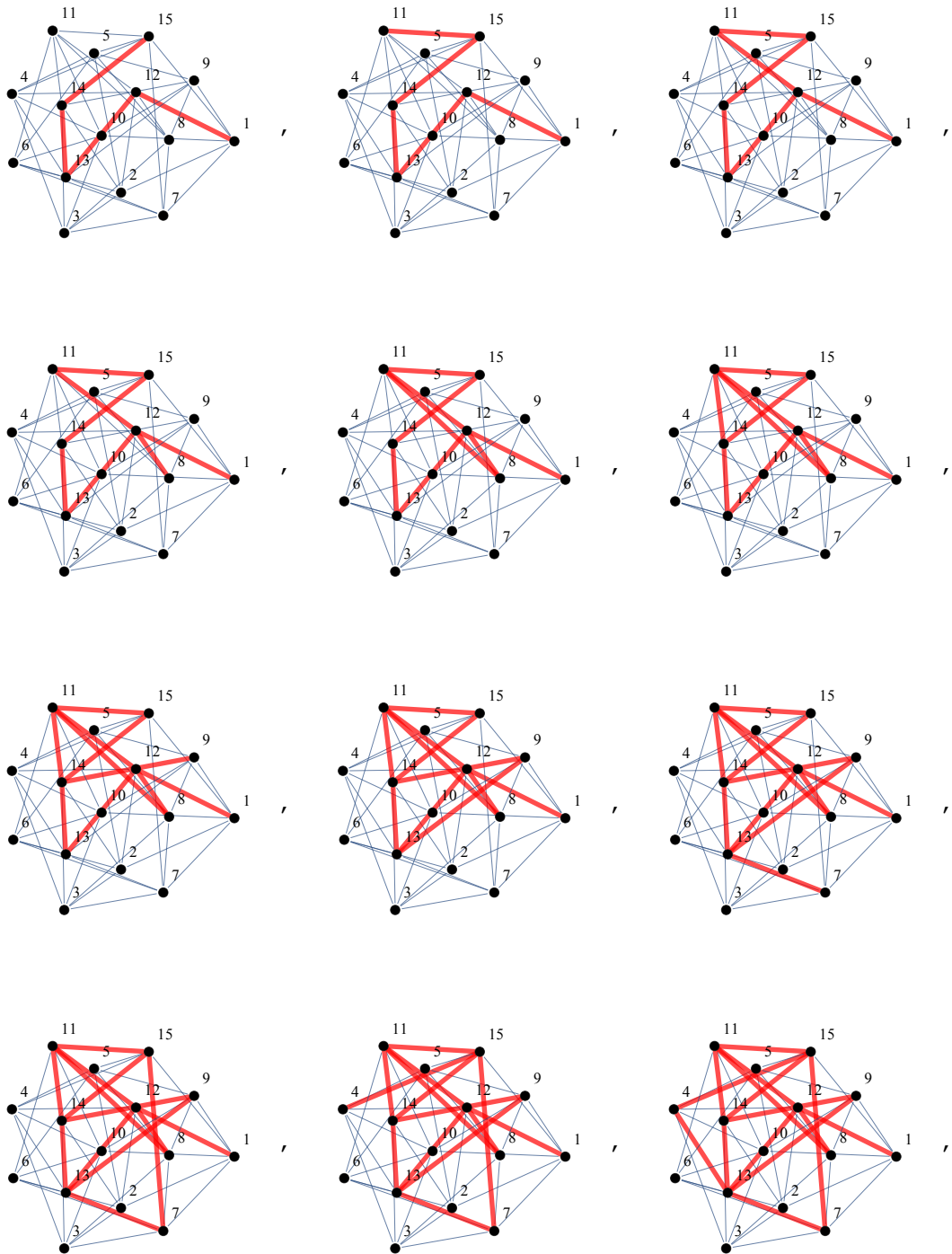


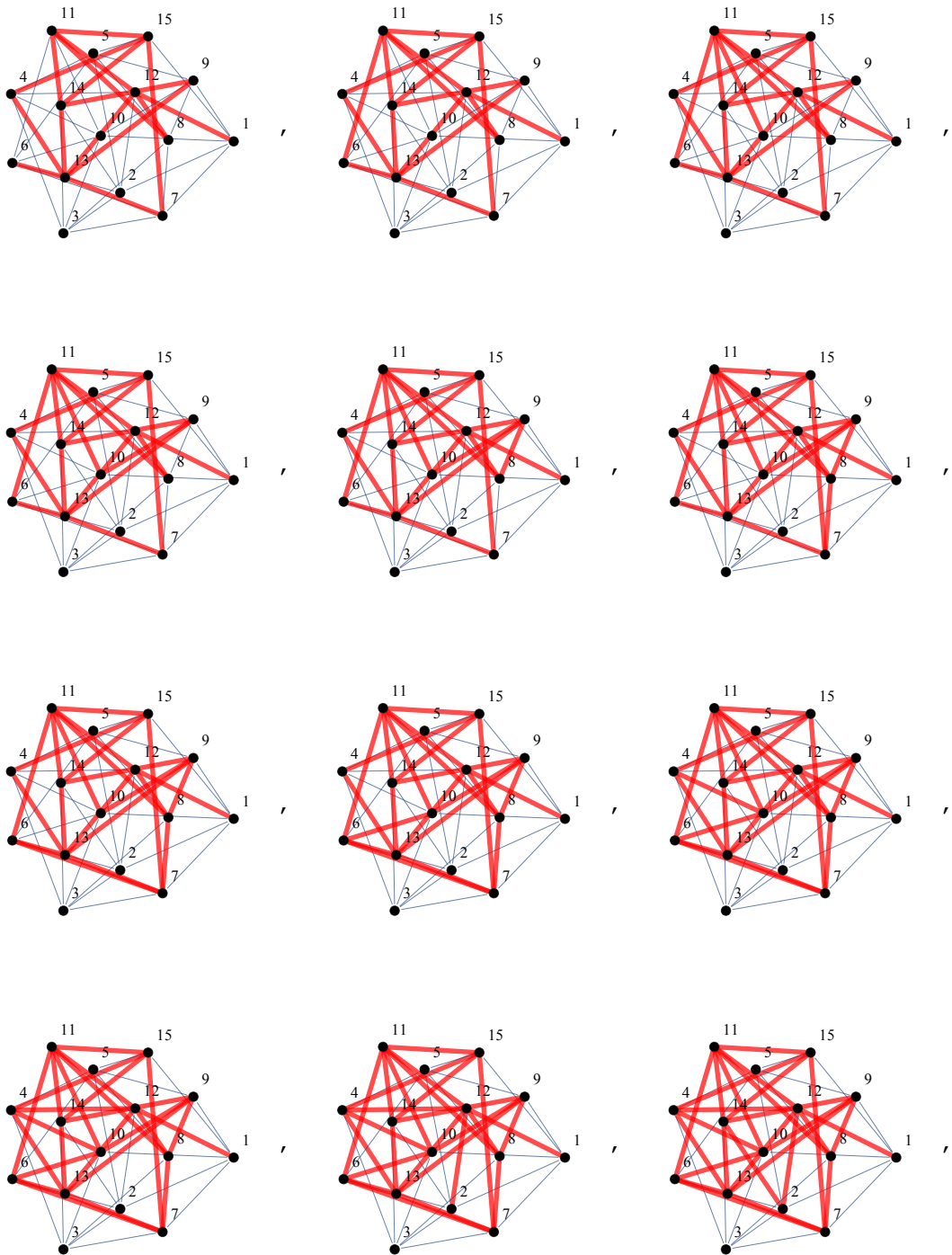
```
eTour = First[FindEulerianCycle[j]]
```

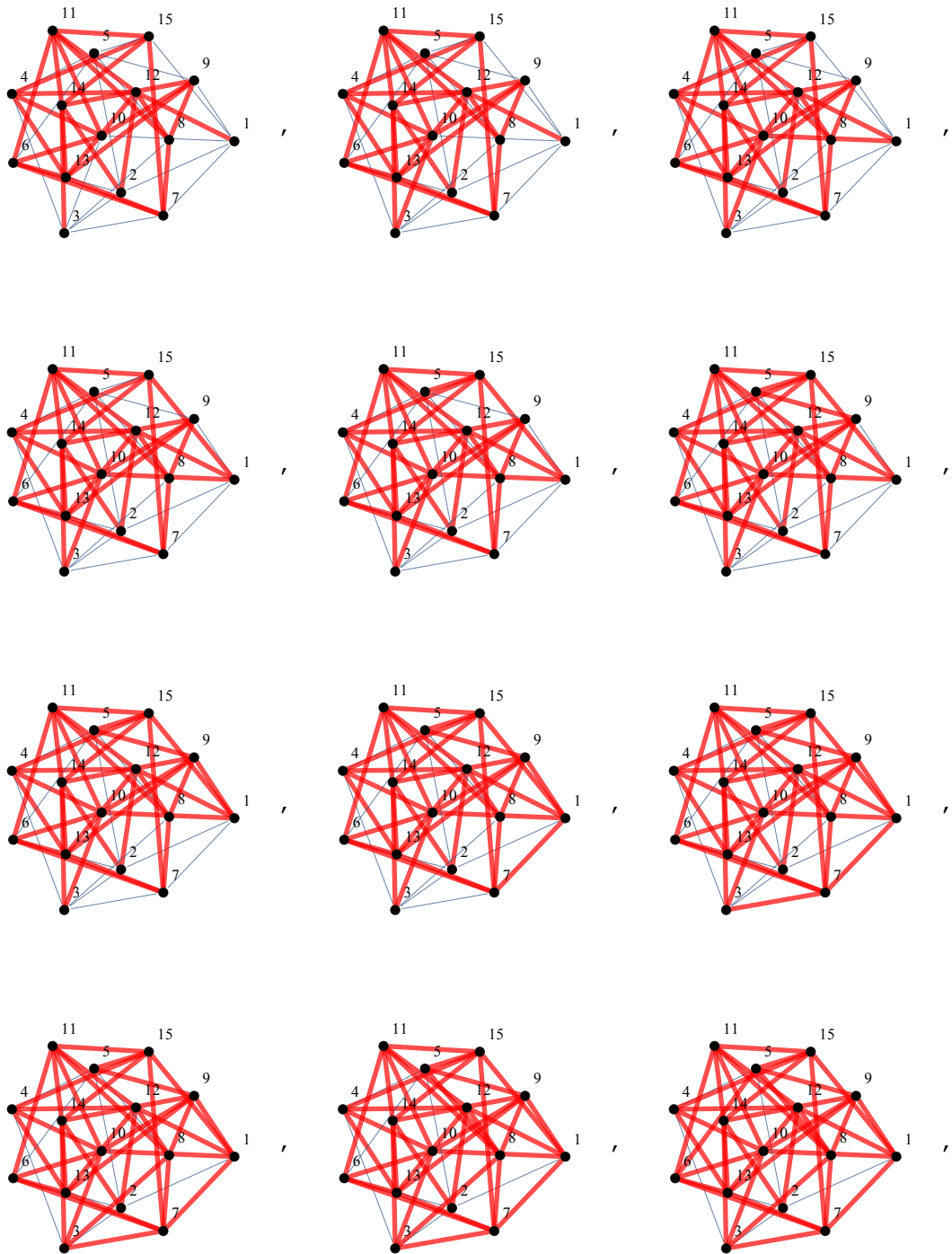
```
{1 ↔ 12, 12 ↔ 13, 13 ↔ 14, 14 ↔ 15, 15 ↔ 11, 11 ↔ 12, 12 ↔ 8, 8 ↔ 11,
  11 ↔ 14, 14 ↔ 9, 9 ↔ 13, 13 ↔ 7, 7 ↔ 15, 15 ↔ 4, 4 ↔ 13, 13 ↔ 6, 6 ↔ 11,
  11 ↔ 10, 10 ↔ 9, 9 ↔ 8, 8 ↔ 7, 7 ↔ 6, 6 ↔ 10, 10 ↔ 4, 4 ↔ 12, 12 ↔ 2,
  2 ↔ 14, 14 ↔ 3, 3 ↔ 10, 10 ↔ 1, 1 ↔ 15, 15 ↔ 5, 5 ↔ 9, 9 ↔ 1, 1 ↔ 7,
  7 ↔ 3, 3 ↔ 8, 8 ↔ 5, 5 ↔ 6, 6 ↔ 2, 2 ↔ 5, 5 ↔ 4, 4 ↔ 3, 3 ↔ 2, 2 ↔ 1}
```

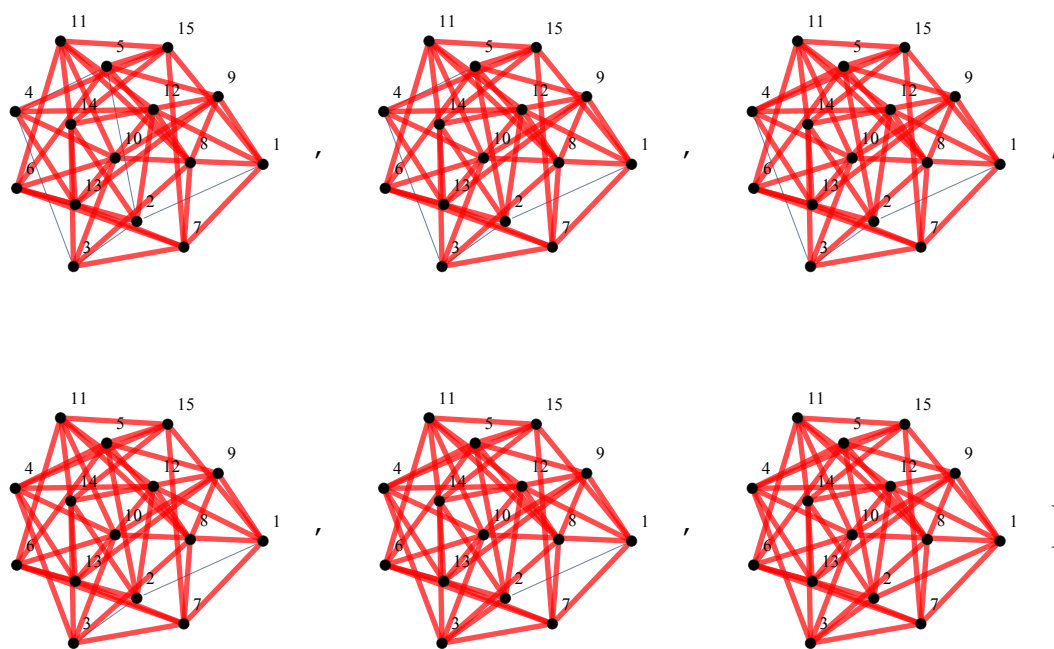
```
Table[HighlightGraph[j, Style[Part[eTour, 1 ;; i], {Thickness[0.02], Red}]],
  {i, Length[eTour]}]
```







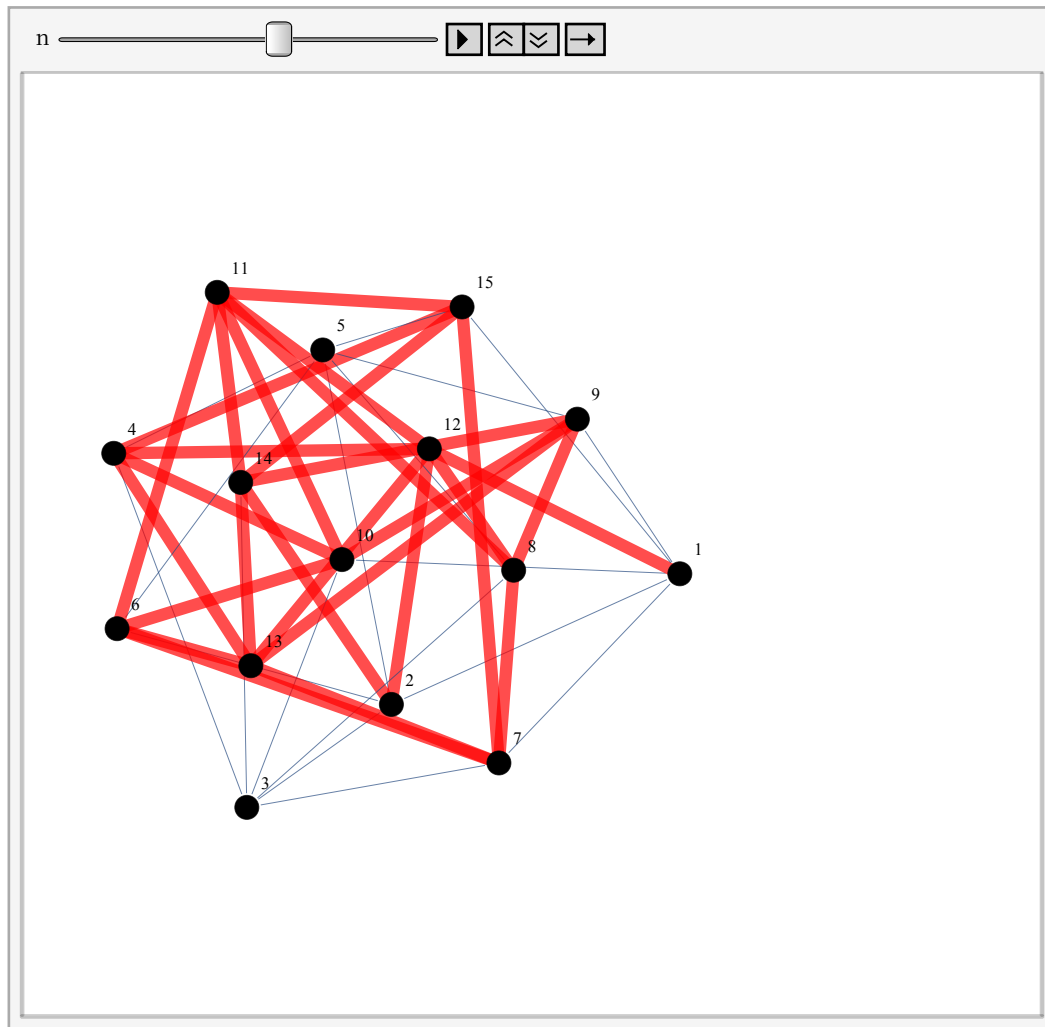




```

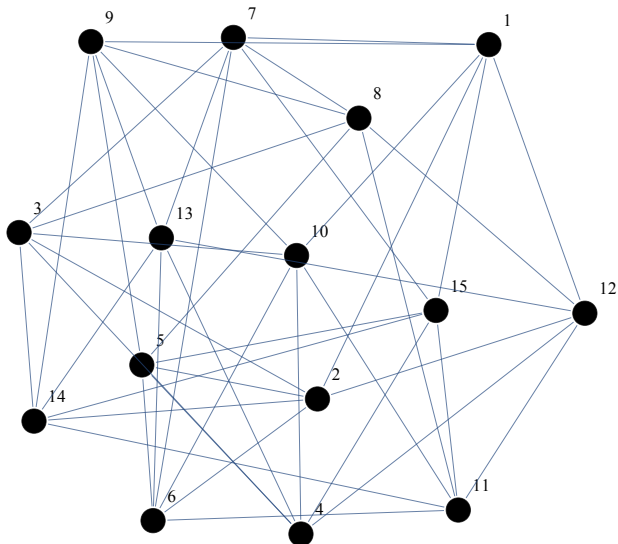
Animate[
  HighlightGraph[j, Style[Take[eTour, n], {Thickness[0.02], Red}]],
  {n, 1, Length[eTour], 1}, AnimationRunning -> False]

```



2. Hamilton cycle

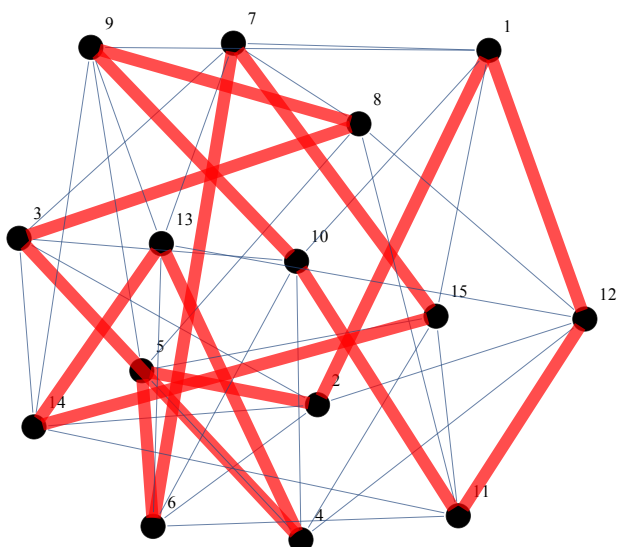
```
g = Graph[Range[7], {1 ↔ 2, 2 ↔ 3, 3 ↔ 4, 4 ↔ 5, 5 ↔ 6, 6 ↔ 7, 7 ↔ 8, 8 ↔ 9, 9 ↔ 10,
  10 ↔ 11, 11 ↔ 12, 12 ↔ 13, 13 ↔ 14, 14 ↔ 15, 15 ↔ 1, 1 ↔ 7, 1 ↔ 9, 1 ↔ 10, 1 ↔ 12,
  2 ↔ 5, 2 ↔ 6, 2 ↔ 12, 2 ↔ 14, 3 ↔ 7, 3 ↔ 8, 3 ↔ 10, 3 ↔ 14, 4 ↔ 10, 4 ↔ 12,
  4 ↔ 13, 4 ↔ 15, 5 ↔ 8, 5 ↔ 9, 5 ↔ 15, 6 ↔ 10, 6 ↔ 11, 6 ↔ 13, 7 ↔ 13, 7 ↔ 15,
  8 ↔ 11, 8 ↔ 12, 9 ↔ 13, 9 ↔ 14, 11 ↔ 14, 11 ↔ 15}, VertexLabels → "Name",
  VertexSize → Medium, VertexStyle → Black, ImagePadding → 20]
```



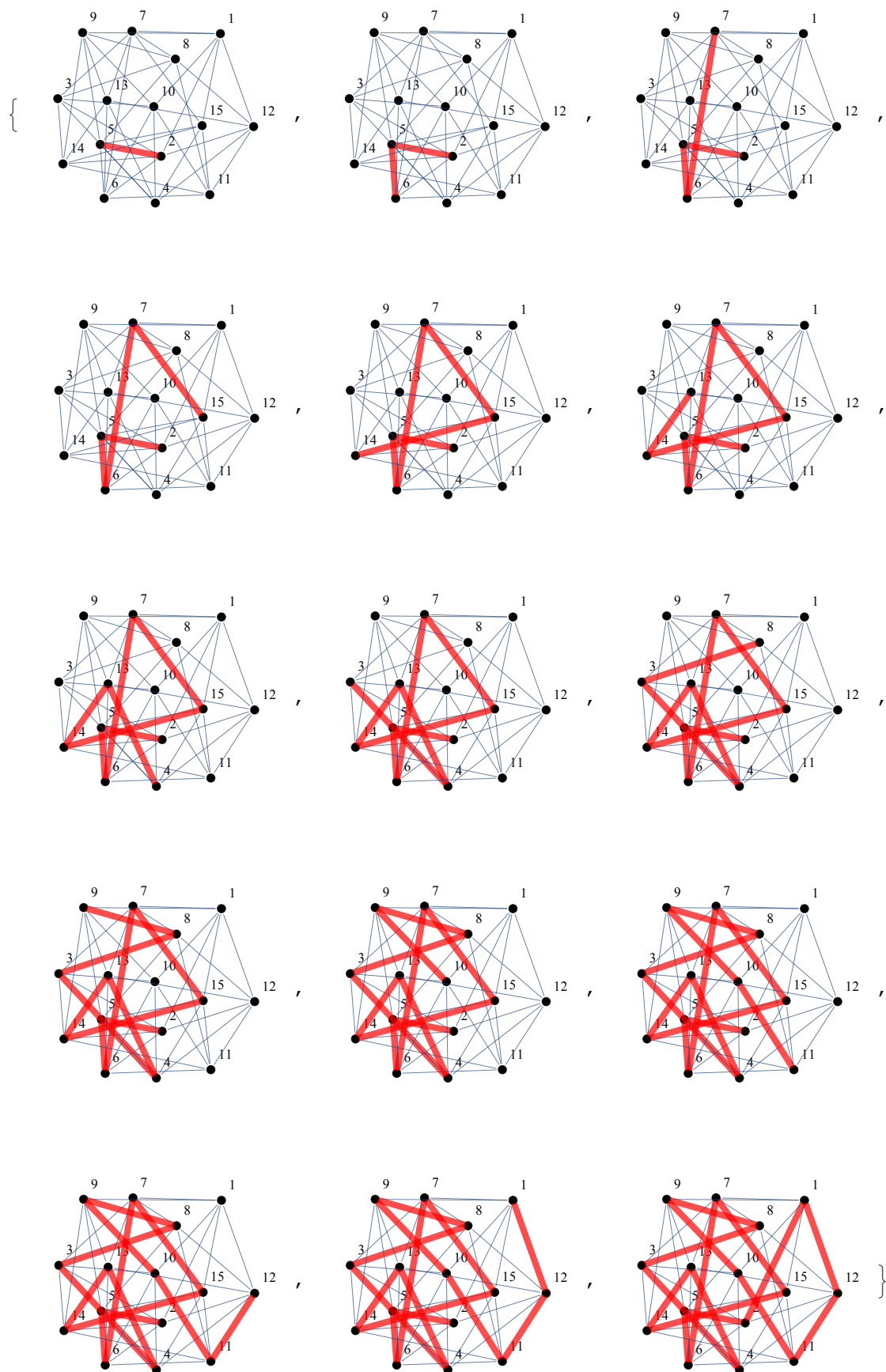
```
hcycle = First[FindHamiltonianCycle[g]]
```

```
{2 ↔ 5, 5 ↔ 6, 6 ↔ 7, 7 ↔ 15, 15 ↔ 14, 14 ↔ 13, 13 ↔ 4,
  4 ↔ 3, 3 ↔ 8, 8 ↔ 9, 9 ↔ 10, 10 ↔ 11, 11 ↔ 12, 12 ↔ 1, 1 ↔ 2}
```

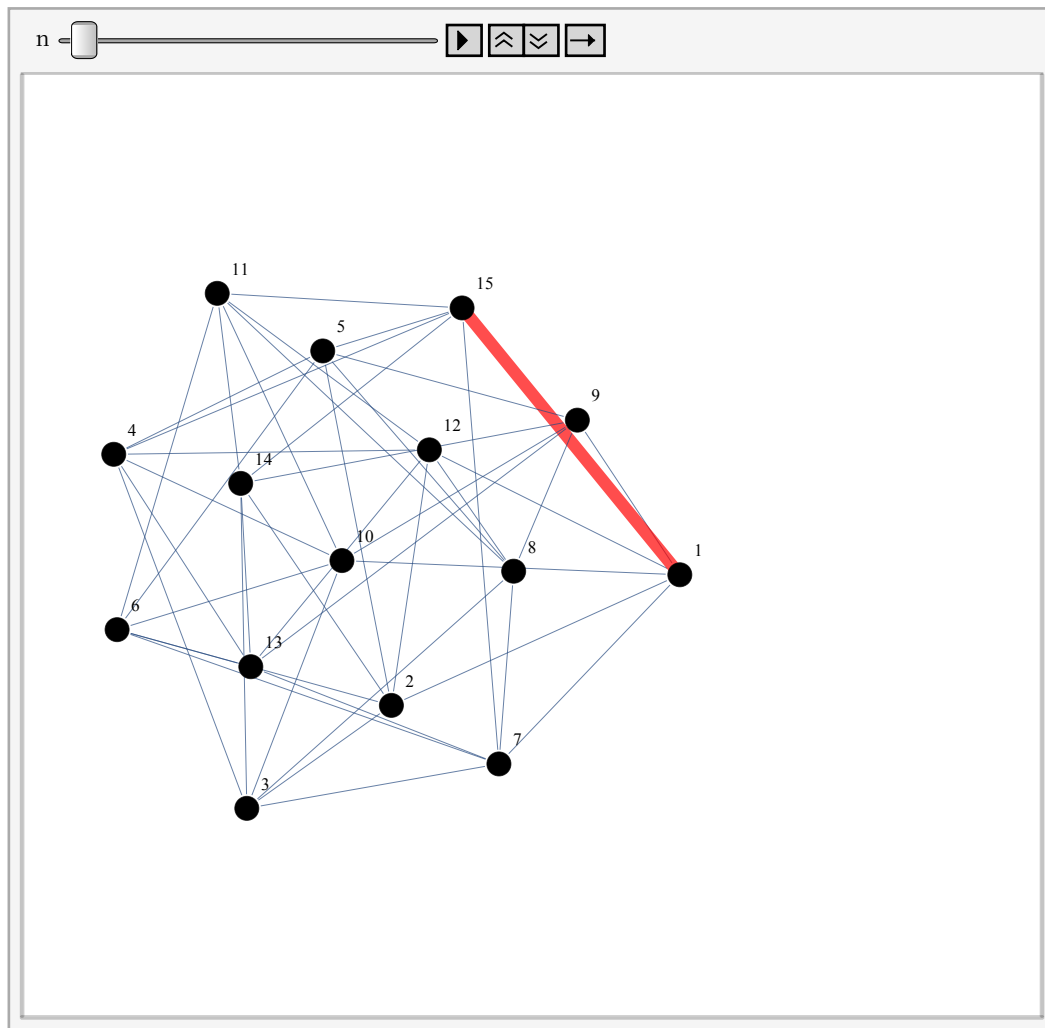
```
HighlightGraph[g, Style[hcycle, {Thickness[0.02], Red}], VertexSize → Medium]
```



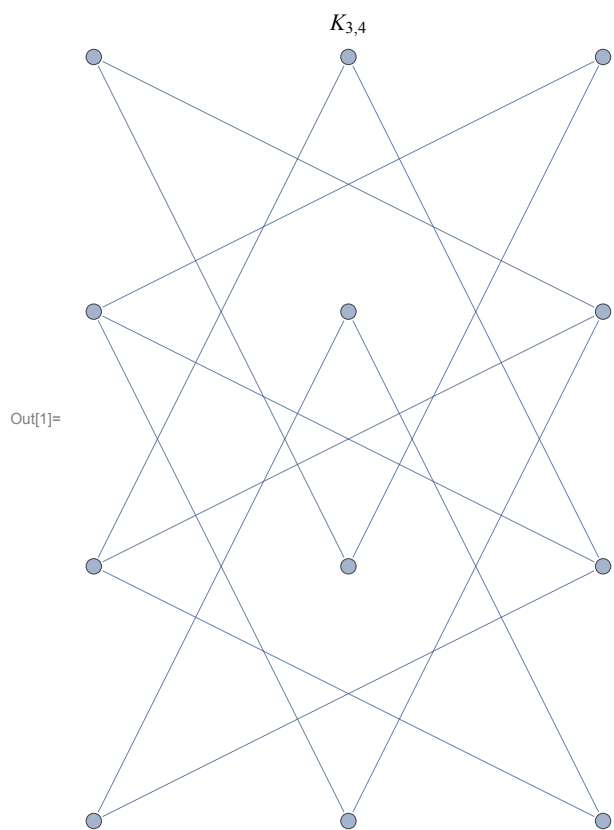
```
HighlightGraph[g, Style[#, {Thickness[0.03], Red}]] & /@  
Table[Take[hcycle, n], {n, 1, Length[hcycle]}]
```



```
Animate[HighlightGraph[g, Style[Take[hcycle, n], {Thickness[0.02], Red}]],  
{n, 1, Length[hcycle], 1}, AnimationRunning -> False]
```



```
In[1]:= a = Table[KnightTourGraph[3, 4, PlotLabel -> K3,4]]
```

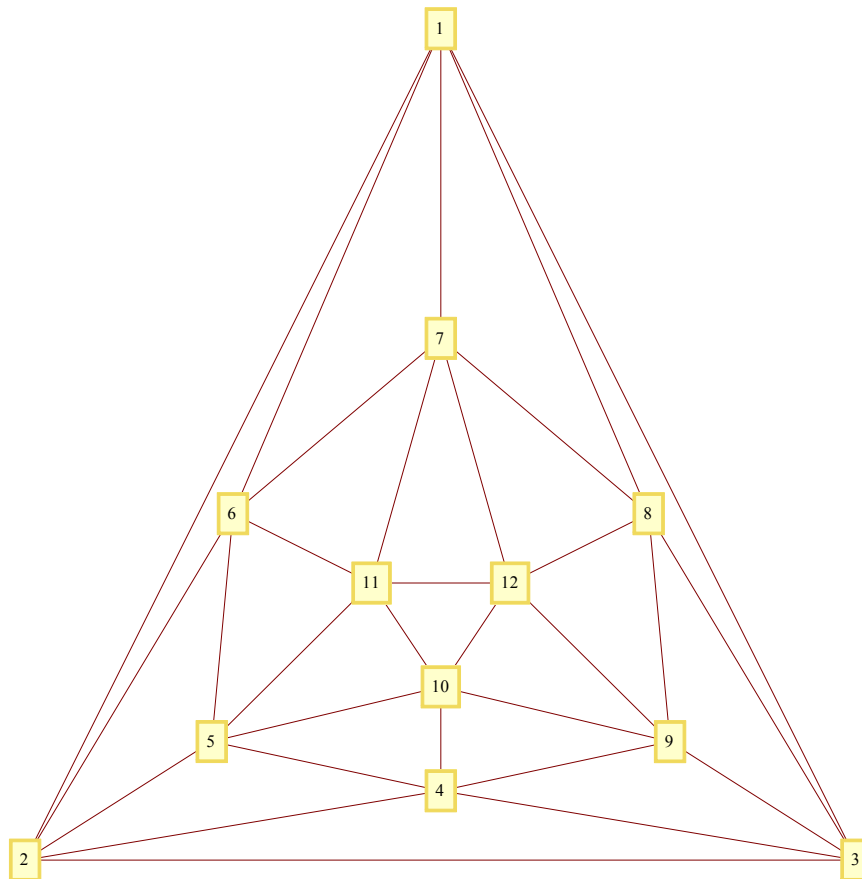


```
FindHamiltonianCycle[a]
```

```
{}
```

7.

```
g = GraphPlot[{1 → 2, 1 → 3, 1 → 6, 1 → 7, 1 → 8, 2 → 3, 2 → 4, 2 → 5, 2 → 6, 3 → 4, 3 → 8,  
  3 → 9, 4 → 5, 4 → 9, 4 → 10, 5 → 6, 5 → 10, 5 → 11, 6 → 7, 6 → 11, 7 → 8, 7 → 11, 7 → 12,  
  8 → 9, 8 → 12, 9 → 10, 9 → 12, 10 → 11, 10 → 12, 11 → 12}, SelfLoopStyle → 0.2,  
  VertexCoordinateRules → {2 → {0, 0}, 3 → {12, 0}, 1 → {6, 12}, 6 → {3, 5}, 8 → {9, 5},  
  4 → {6, 1}, 10 → {6, 2.5}, 11 → {5, 4}, 12 → {7, 4}}, VertexLabeling → True]
```

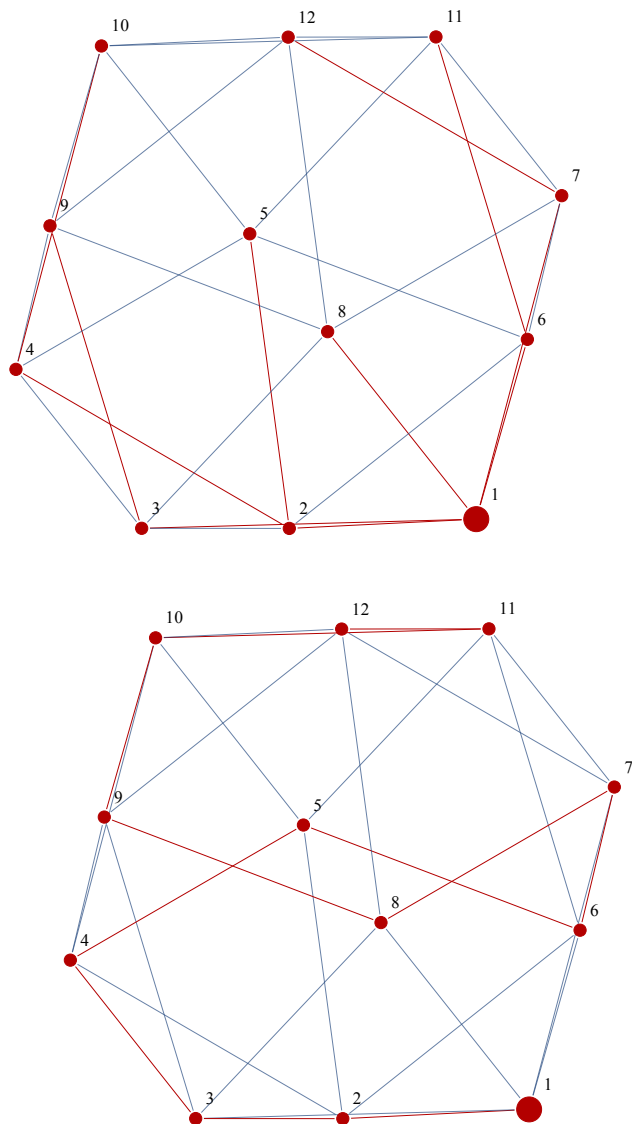


```

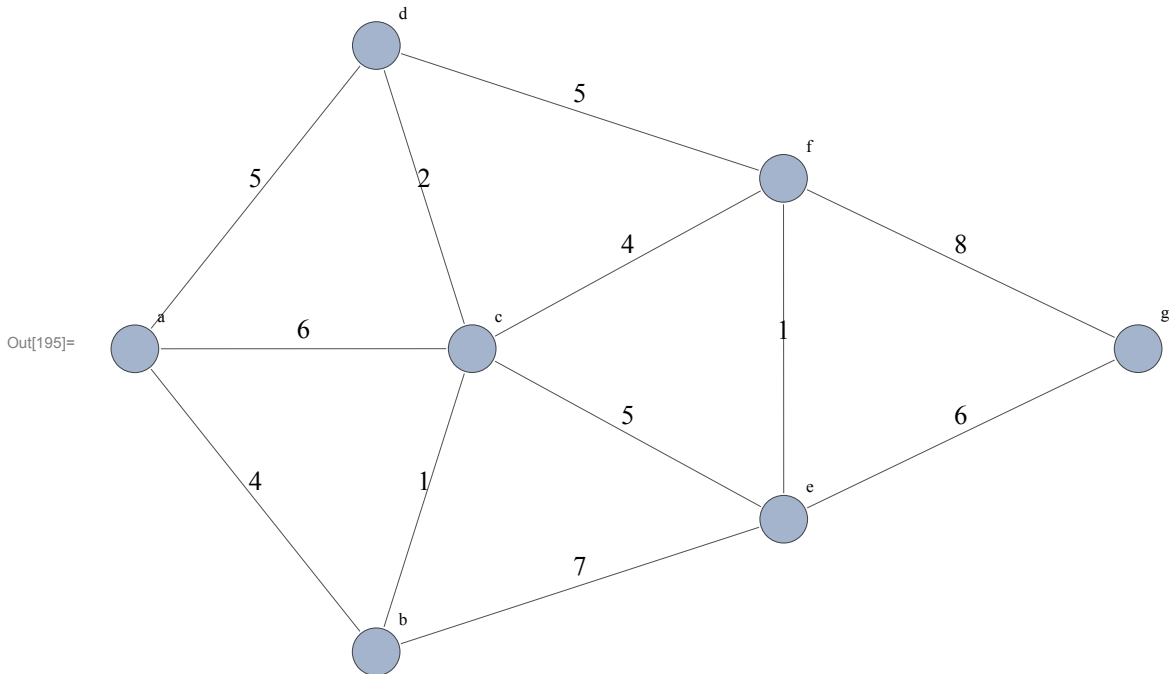
In[1]:= g = Graph[Range[12], {1 -> 2, 1 -> 3, 1 -> 6, 1 -> 7, 1 -> 8, 2 -> 3,
  2 -> 4, 2 -> 5, 2 -> 6, 3 -> 4, 3 -> 8, 3 -> 9, 4 -> 5, 4 -> 9, 4 -> 10, 5 -> 6,
  5 -> 10, 5 -> 11, 6 -> 7, 6 -> 11, 7 -> 8, 7 -> 11, 7 -> 12, 8 -> 9, 8 -> 12,
  9 -> 10, 9 -> 12, 10 -> 11, 10 -> 12, 11 -> 12}, VertexLabels -> "Name"];
Row[{HighlightGraph[g, {TreeGraph[VertexList[g], BreadthFirstScan[g, 1]}],
  VertexSize -> {1 -> 0.2}], Spacer[20], HighlightGraph[g,
  TreeGraph[VertexList[g], DepthFirstScan[g, 1]], VertexSize -> {1 -> 0.2}]]]

```

Out[2]=



```
Graph[{"a" ê "b", "a" ê "c", "a" ê "d", "b" ê "c",
      "b" ê "e", "c" ê "d", "c" ê "e", "c" ê "f", "d" ê "f",
      "f" ê "e", "e" ê "g", "f" ê "g"}, VertexLabels → "Name",
      EdgeWeight → {4, 6, 5, 1, 7, 2, 5, 4, 5, 1, 6, 8}, VertexSize → 0.15,
      EdgeStyle → Black, EdgeLabels → "EdgeWeight", EdgeLabelStyle → 14]
```



```
mydata = {{0, 4, 6, 5, Infinity, Infinity, Infinity},
          {4, 0, 1, Infinity, 7, Infinity, Infinity},
          {6, 1, 0, 2, 5, 4, Infinity}, {5, Infinity, 2, 0, Infinity, 5, Infinity},
          {Infinity, 7, 5, Infinity, 0, 6}, {Infinity, Infinity, 4, 5, 1, 0, 8},
          {Infinity, Infinity, Infinity, Infinity, 6, 8, 0}};
mygraph = FromAdjacencyMatrix[mydata, EdgeWeight];
GraphPlot[mygraph, VertexLabeling → True];

A = ToAdjacencyMatrix[FromAdjacencyMatrix[mydata, EdgeWeight]];
B = DeleteCases[Flatten[Table[If[And[i ≠ j, A[[i, j]] = a],
    {j → i, ToString[mydata[[i, j]]}], {i, g}, {j, i}], a], Null];

GraphPlot[B, VertexLabeling → True];
aaa = MinimumSpanningTree[mygraph];
ans = GraphPlot[aaa, VertexLabeling → True];

xpts = Cases[ans, Line[pts : _] => pts, Infinity][[a]];
vertex = VertexRenderingFunction →
  ({EdgeForm[Black], Yellow, Disk[#1, 0.1], Black, Text[#2, #1]} &);
Show[GraphPlot[mygraph, VertexLabeling → True,
  PlotStyle → {Blue, Thickness[0.01]}, vertex] /. Line[_] => Line[xpts],
  GraphPlot[B, VertexLabeling → True, vertex]]
```