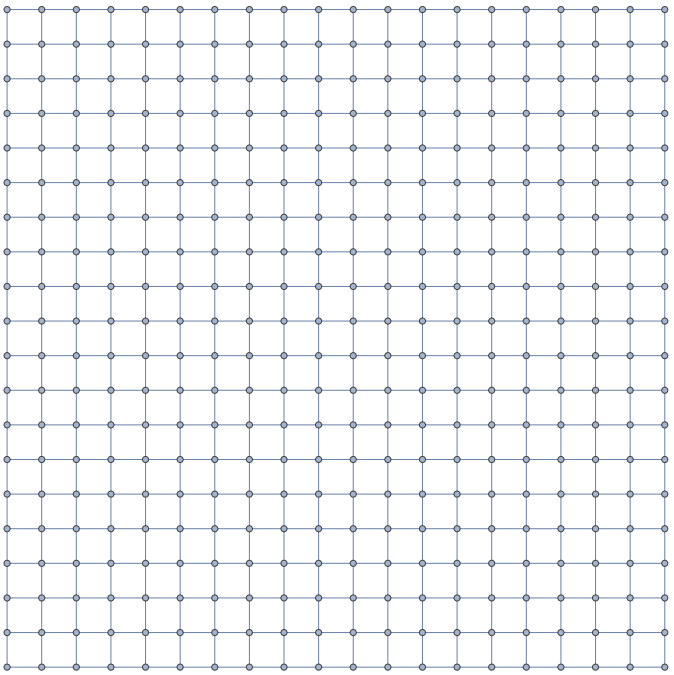
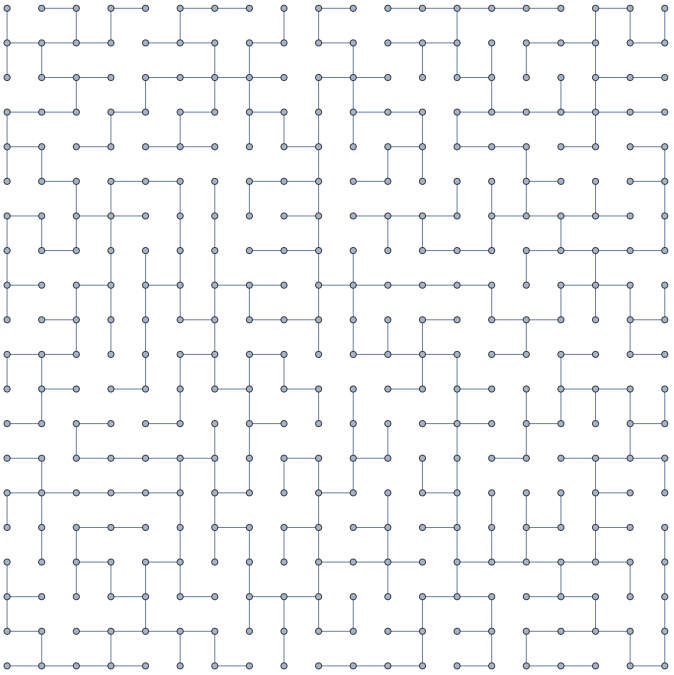
In[ ]:=
```
g = GridGraph[{20, 20}, EdgeWeight → RandomInteger[10, 760]]
```
网格图　边的权值　伪随机整数

Out[ ]=



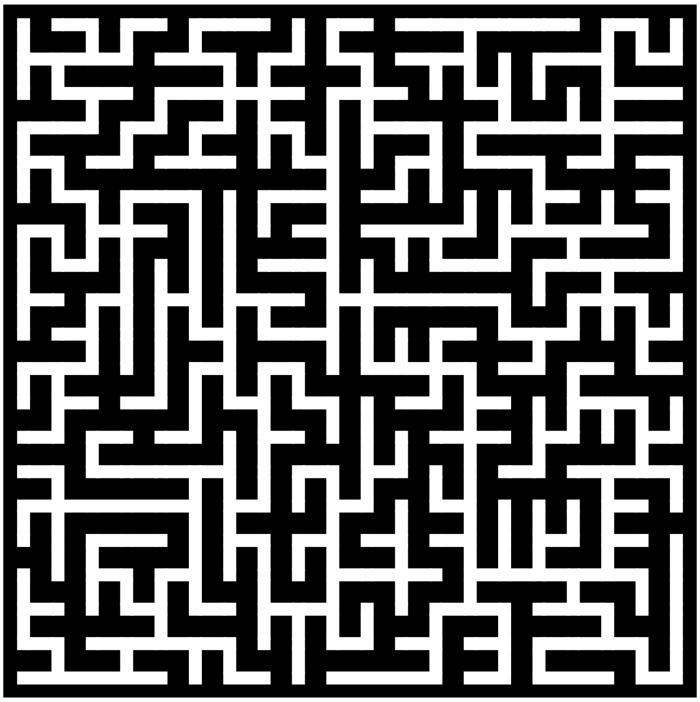
In[ ]:=
```
tree = FindSpanningTree[g]
```
找到生成树

Out[ ]=



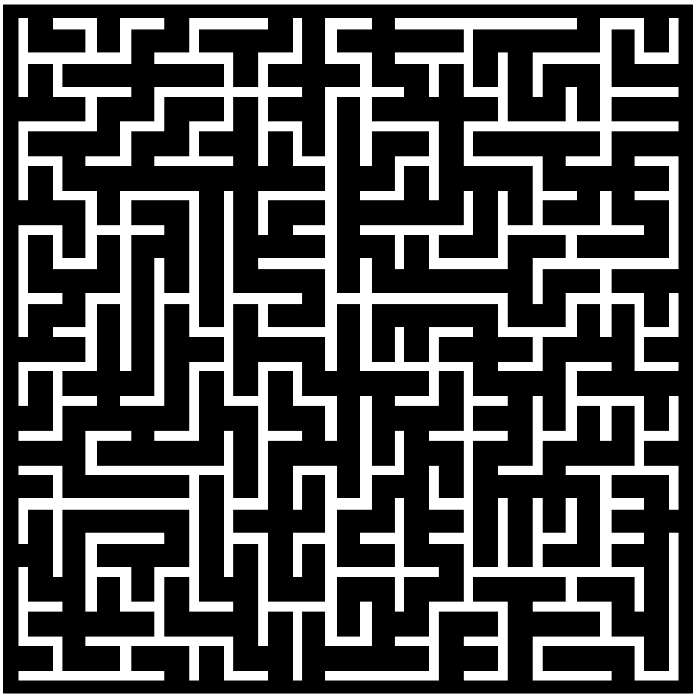
In[ ]:=
```
maze = Graph[tree, Background → Black, BaseStyle → {White, Opacity[1], EdgeForm[]},
        图              背景色           黑色      基本样式        白色    不透明度     边的格式
  EdgeShapeFunction → (Rectangle[#1[[1]] + 0.15, #1[[2]] - 0.15] &),
  边的形状函数              矩形
  VertexShapeFunction → (Rectangle[#1 + 0.15, #1 - 0.15] &)]
  顶点形状函数                矩形
```

Out[ ]=



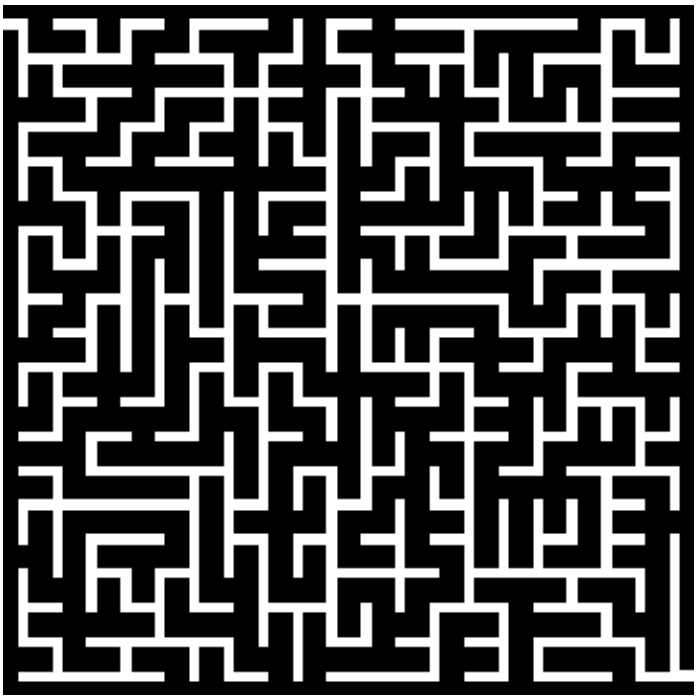
In[ ]:=
```
img = Image[maze]
      图像
```

Out[ ]=



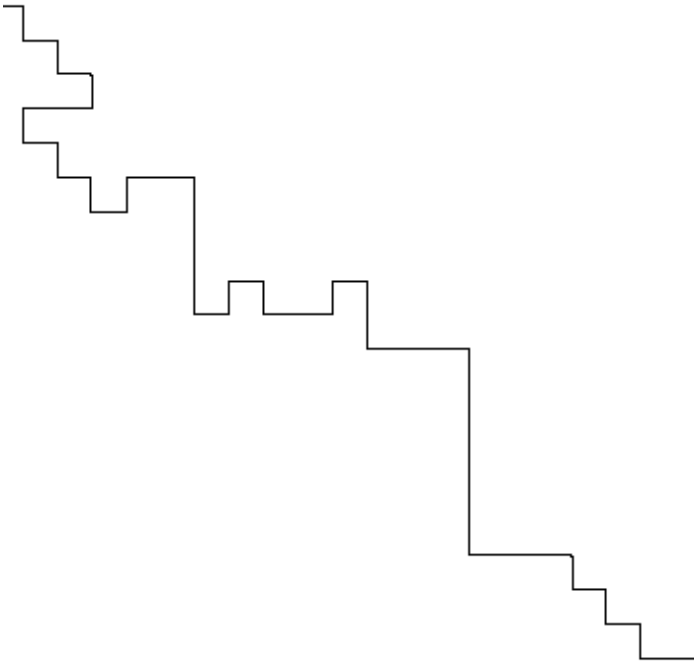
In[ ]:=
```
pos[x_, y_] := Flatten[Table[{i + x, j + y} → White, {i, 7}, {j, 5}]]
                压平      表格                白色
img = ReplaceImageValue[img, pos[352, 7] ~ Join ~ pos[0, 346]]
      替换图像值                                  连接
```

Out[ ]=



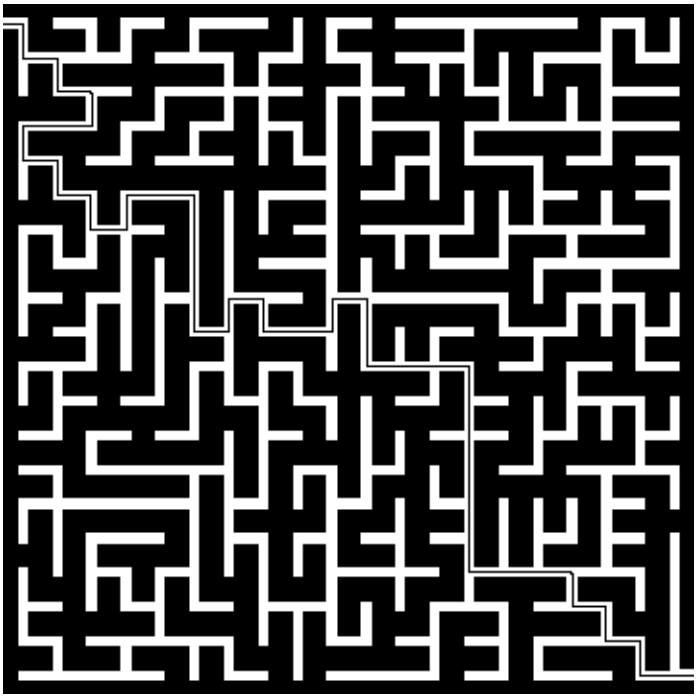
In[ ]:=
```
path = Image[WatershedComponents[img], "Bit"]
       图像    分水岭分量
```

Out[ ]=



In[ ]:=
```
ImageMultiply[img, path]
```
图像相乘

Out[ ]=



In[ ]:=
```
ReplaceImageValue[img, # → Red & /@ ImageValuePositions[path, Black]]
```
替换图像值 红色 图像值位置 黑色

Out[ ]=