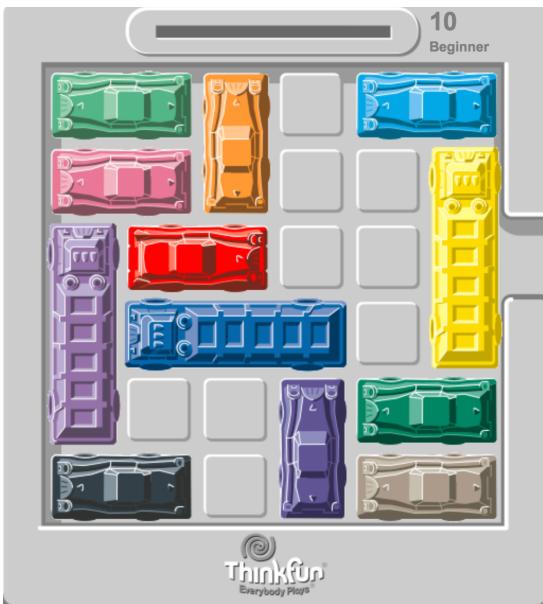


```

cars = {<|"Positions" → {{1, 1}, {1, 2}}, "Directions" → {{0, 1}, {0, -1}}|>, <|
    "Positions" → {{3, 2}, {3, 3}, {3, 4}}, "Directions" → {{0, 1}, {0, -1}}|>, <|
    "Positions" → {{2, 1}, {2, 2}, {2, 3}}, "Directions" → {{0, -1}, {0, 1}}|>, <|
    "Positions" → {{1, 5}, {2, 5}, {3, 5}}, "Directions" → {{-1, 0}, {1, 0}}|>};


```



```

cars={

    <|"Positions"→{{2,4},{3,4}),"Directions"→{{-1,0},{1,0}}|>,
    <|"Positions"→{{2,3},{3,3},{4,3}),"Directions"→{{-1,0},{1,0}}|>,
    <|"Positions"→{{1,1},{2,1}),"Directions"→{{-1,0},{1,0}}|>,
    <|"Positions"→{{1,5},{2,5}),"Directions"→{{-1,0},{1,0}}|>,
    <|"Positions"→{{1,6},{2,6}),"Directions"→{{-1,0},{1,0}}|>,
    <|"Positions"→{{5,6},{6,6}),"Directions"→{{-1,0},{1,0}}|>,
    <|"Positions"→{{5,1},{6,1}),"Directions"→{{-1,0},{1,0}}|>,
    <|"Positions"→{{5,2},{6,2}),"Directions"→{{-1,0},{1,0}}|>,
    <|"Positions"→{{1,2},{1,3},{1,4}),"Directions"→{{0,-1},{0,1}}|>,
    <|"Positions"→{{6,3},{6,4},{6,5}),"Directions"→{{0,-1},{0,1}}|>,
    <|"Positions"→{{4,1},{4,2}),"Directions"→{{0,-1},{0,1}}|>,
    <|"Positions"→{{3,5},{3,6}),"Directions"→{{0,-1},{0,1}}|>

};


```

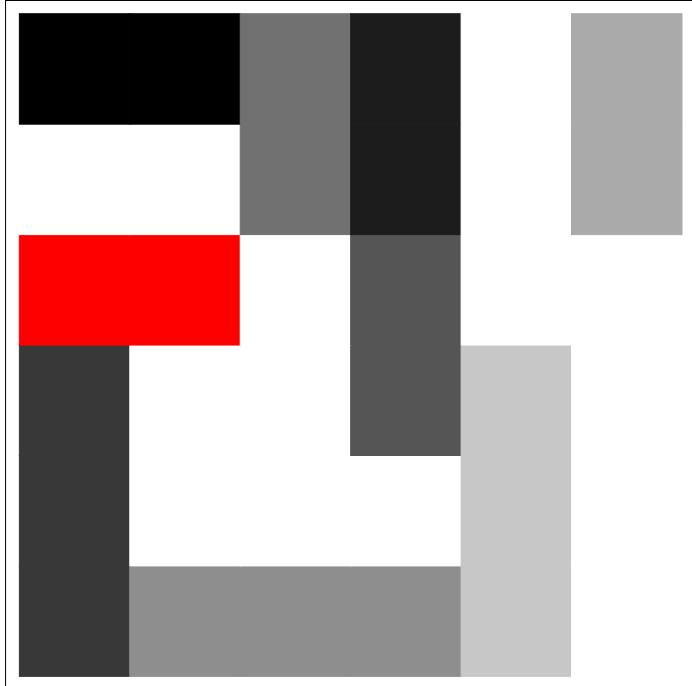
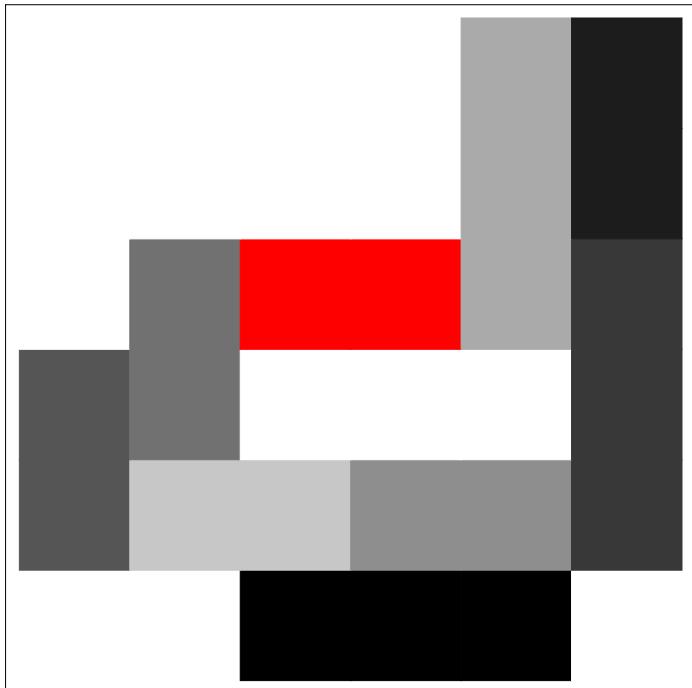
```

randomBoard[n_, boardDims_, initialCars_: {}]:=

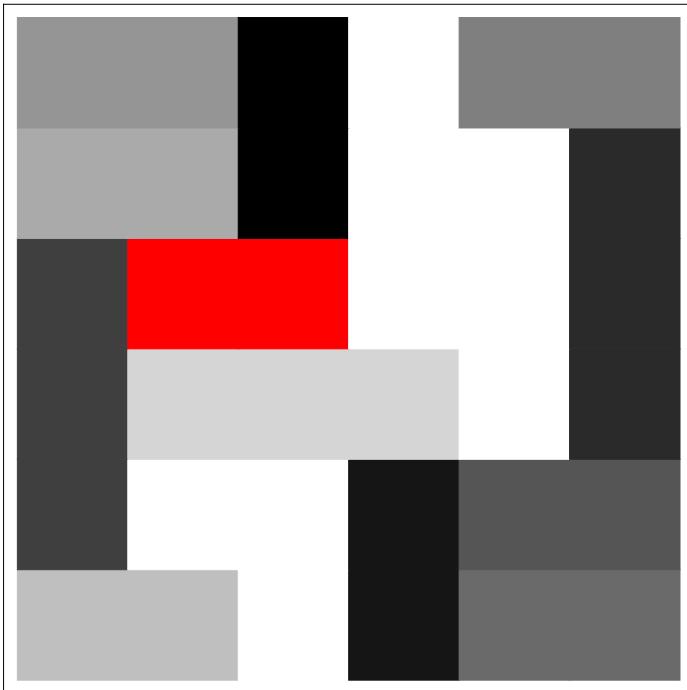
NestWhile[Join[initialCars, Table[With[
    {b = RandomInteger[{1, #}] & /@boardDims}, RandomChoice[{<|"Positions" →
        {b, b + {0, 1}}, RandomChoice[Unevaluated@{Nothing, b + {0, 2}}]|>,
        "Directions" → {{0, -1}, {0, 1}}|>, <|"Positions" →
        {b, b + {1, 0}}, RandomChoice[Unevaluated@{Nothing, b + {2, 0}}]|>},
        "Directions" → {{-1, 0}, {1, 0}}|>}], n] &,
    Missing[], ! validBoard[#, boardDims] &]


```

```
showBoard[cars = randomBoard[8, {6, 6},  
  {<|"Positions" → {{#, 4}, {#+1, 4}} &@RandomInteger[{1, 6 - 1}],  
   "Directions" → {{-1, 0}, {1, 0}}|>}, {6, 6}]
```

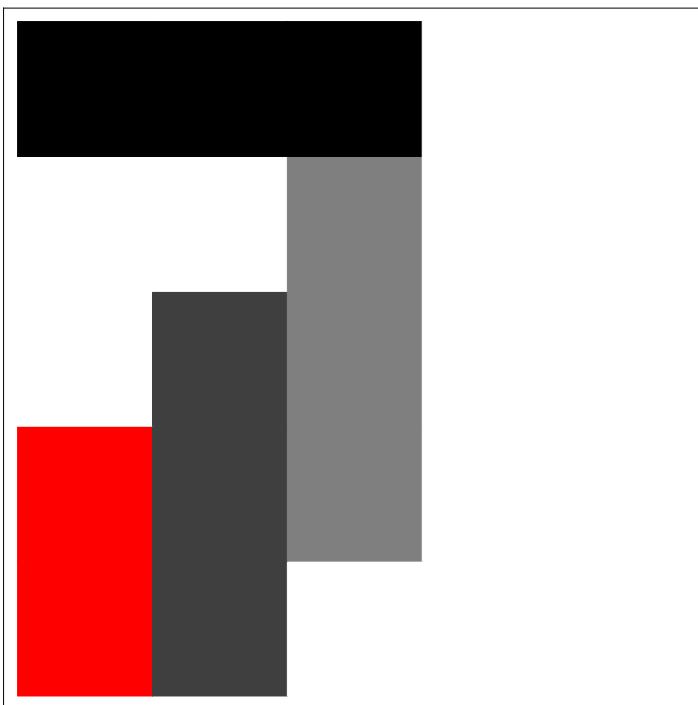


```
showBoard[cars, {6, 6}]
```



```
showBoard[cars_, boardDims_] := ArrayPlot[Reverse@Transpose@  
SparseArray[Flatten@MapIndexed[(n \[Rule] n \[Rule] First@#2) /@ #1["Positions"] &, cars],  
boardDims], ColorRules \[Rule] {1 \[Rule] Red}]
```

```
showBoard[cars, {5, 5}]
```



```

validBoard[cars : {_Association}, boardDims_] :=
  With[{pos = Catenate@cars[[All, "Positions"]]},
    DuplicateFreeQ[pos] && AllTrue[pos[[All, 1]], Between[{1, boardDims[[1]]}]] &&
    AllTrue[pos[[All, 2]], Between[{1, boardDims[[2]]}]]]

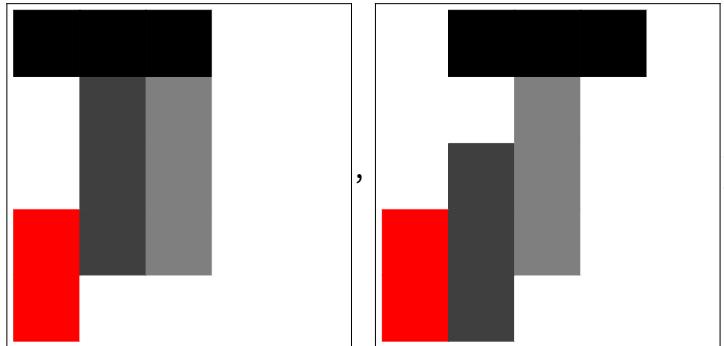
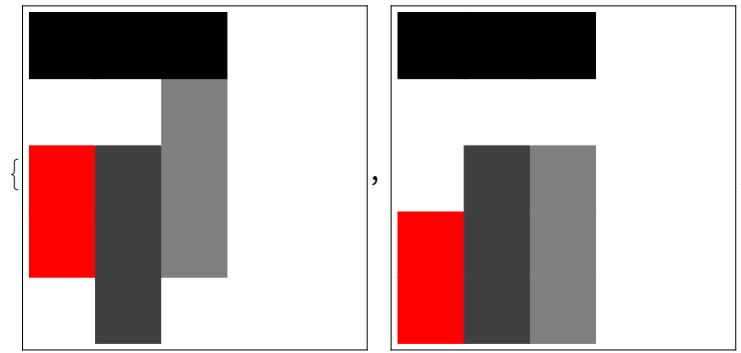
validBoard[_, _] := False

singleMoves[cars_, boardDims_] := Select[Catenate@MapIndexed[{(c, i) \[Function]
  ReplacePart[cars, i \[Function] <|"Positions" \[Function] #, "Directions" \[Function] c["Directions"]|>] &/@
    Table[pos + dir, {dir, c["Directions"]}], {pos, c["Positions"]}]], cars], validBoard[#, boardDims] &]

singleMoves[cars_, boardDims_] := Select[Catenate@MapIndexed[{(c, i) \[Function]
  ReplacePart[cars, i \[Function] <|"Positions" \[Function] #, "Directions" \[Function] c["Directions"]|>] &/@
    Table[pos + dir, {dir, c["Directions"]}], {pos, c["Positions"]}]], cars],
  With[{pos = Catenate@#[[All, "Positions"]]}, DuplicateFreeQ[pos] &&
    AllTrue[pos[[All, 1]], Between[{1, boardDims[[1]]}]] &&
    AllTrue[pos[[All, 2]], Between[{1, boardDims[[2]]}]]] &]

showBoard[#, {5, 5}] &/@singleMoves[cars, {5, 5}]

```



```

VertexList@

NestGraph[DeleteDuplicates@*Catenate@*Map[singleMoves[#, {5, 5}] &], {cars}, 15]

{{<| Positions → {{1, 1}, {1, 2}}, Directions → {{0, 1}, {0, -1}}|>,
  <| Positions → {{3, 2}, {3, 3}, {3, 4}}, Directions → {{0, 1}, {0, -1}}|>,
  <| Positions → {{2, 1}, {2, 2}, {2, 3}}, Directions → {{0, -1}, {0, 1}}|>,
  <| Positions → {{1, 5}, {2, 5}, {3, 5}}, Directions → {{-1, 0}, {1, 0}}|>}, {}}

cars

{<| Positions → {{1, 1}, {1, 2}}, Directions → {{0, 1}, {0, -1}}|>,
  <| Positions → {{3, 2}, {3, 3}, {3, 4}}, Directions → {{0, 1}, {0, -1}}|>,
  <| Positions → {{2, 1}, {2, 2}, {2, 3}}, Directions → {{0, -1}, {0, 1}}|>,
  <| Positions → {{1, 5}, {2, 5}, {3, 5}}, Directions → {{-1, 0}, {1, 0}}|>}

showBoard[#, {5, 5}] & /@DeleteDuplicates@Catenate@

NestList[DeleteDuplicates@*Catenate@*Map[singleMoves[#, {5, 5}] &], {cars}, 15]

Block[{pos = <|Missing[] → cars|>}, Merge[
  {pos, # → singleMoves[#, {5, 5}] & /@Complement[Values@pos, Keys@pos]}, First]]

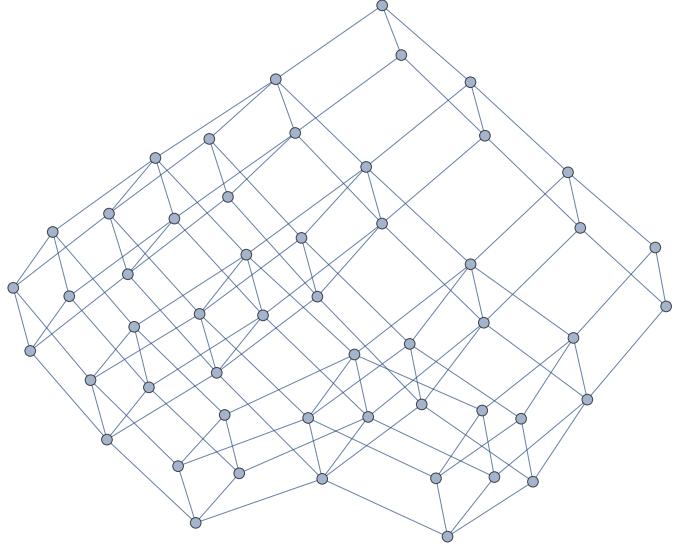
<|Missing[] → {<| Positions → {{1, 1}, {1, 2}}, Directions → {{0, 1}, {0, -1}}|>,
  <| Positions → {{3, 2}, {3, 3}, {3, 4}}, Directions → {{0, 1}, {0, -1}}|>,
  <| Positions → {{2, 1}, {2, 2}, {2, 3}}, Directions → {{0, -1}, {0, 1}}|>,
  <| Positions → {{1, 5}, {2, 5}, {3, 5}}, Directions → {{-1, 0}, {1, 0}}|>},
 {<| Positions → {{1, 1}, {1, 2}}, Directions → {{0, 1}, {0, -1}}|>,
  <| Positions → {{3, 2}, {3, 3}, {3, 4}}, Directions → {{0, 1}, {0, -1}}|>,
  <| Positions → {{2, 1}, {2, 2}, {2, 3}}, Directions → {{0, -1}, {0, 1}}|>,
  <| Positions → {{1, 5}, {2, 5}, {3, 5}}, Directions → {{-1, 0}, {1, 0}}|>} →
  {{<| Positions → {{1, 2}, {1, 3}}, Directions → {{0, 1}, {0, -1}}|>,
    <| Positions → {{3, 2}, {3, 3}, {3, 4}}, Directions → {{0, 1}, {0, -1}}|>,
    <| Positions → {{2, 1}, {2, 2}, {2, 3}}, Directions → {{0, -1}, {0, 1}}|>,
    <| Positions → {{1, 5}, {2, 5}, {3, 5}}, Directions → {{-1, 0}, {1, 0}}|>},
   {<| Positions → {{1, 1}, {1, 2}}, Directions → {{0, 1}, {0, -1}}|>,
    <| Positions → {{3, 1}, {3, 2}, {3, 3}}, Directions → {{0, 1}, {0, -1}}|>,
    <| Positions → {{2, 1}, {2, 2}, {2, 3}}, Directions → {{0, -1}, {0, 1}}|>,
    <| Positions → {{1, 5}, {2, 5}, {3, 5}}, Directions → {{-1, 0}, {1, 0}}|>},
   {<| Positions → {{1, 1}, {1, 2}}, Directions → {{0, 1}, {0, -1}}|>,
    <| Positions → {{3, 2}, {3, 3}, {3, 4}}, Directions → {{0, 1}, {0, -1}}|>,
    <| Positions → {{2, 2}, {2, 3}, {2, 4}}, Directions → {{0, -1}, {0, 1}}|>,
    <| Positions → {{1, 5}, {2, 5}, {3, 5}}, Directions → {{-1, 0}, {1, 0}}|>},
   {<| Positions → {{1, 1}, {1, 2}}, Directions → {{0, 1}, {0, -1}}|>,
    <| Positions → {{3, 2}, {3, 3}, {3, 4}}, Directions → {{0, 1}, {0, -1}}|>,
    <| Positions → {{2, 1}, {2, 2}, {2, 3}}, Directions → {{0, -1}, {0, 1}}|>,
    <| Positions → {{2, 5}, {3, 5}, {4, 5}}, Directions → {{-1, 0}, {1, 0}}|>}}|>

```

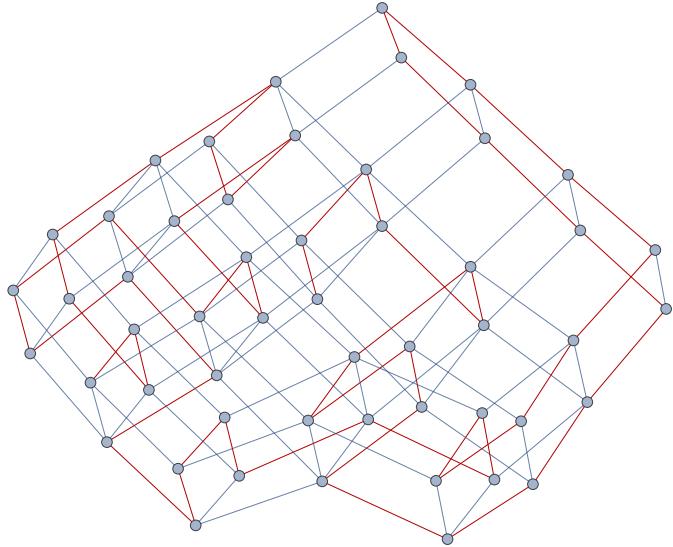
```

g = Graph@
  DeleteDuplicatesBy[UndirectedEdge @@@ Rest@FixedPoint[posss & Catenate[{posss,
    Catenate[(pos & pos -> # & /@ singleMoves[pos, {5, 5}]) /@ Complement[
      Values@posss, Keys@posss]]}], {Missing[] & cars}], Sort]

```



```
HighlightGraph[g, PathGraph@FindHamiltonianPath[g]]
```



<http://www.ulb.ac.be/di/algo/secollet/papers/crs06.pdf>

```

g = Graph@
  DeleteDuplicatesBy[UndirectedEdge @@@ Rest@FixedPoint[posss & Catenate[{posss,
    Catenate[(pos & pos -> # & /@ singleMoves[pos, {6, 6}]) /@ Complement[
      Values@posss, Keys@posss]]}], {Missing[] & cars}], Sort];

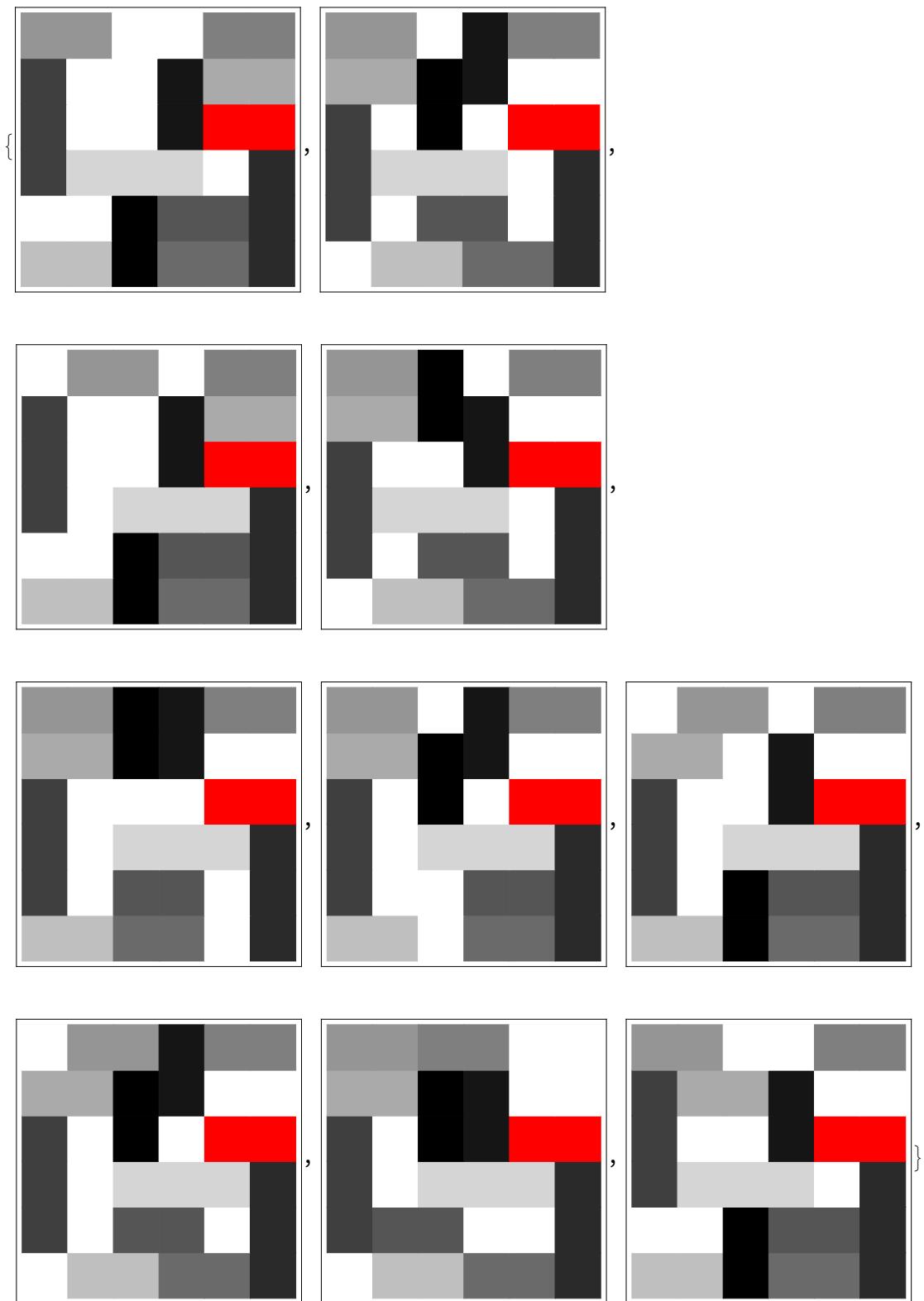
```

```
VertexCount@g
```

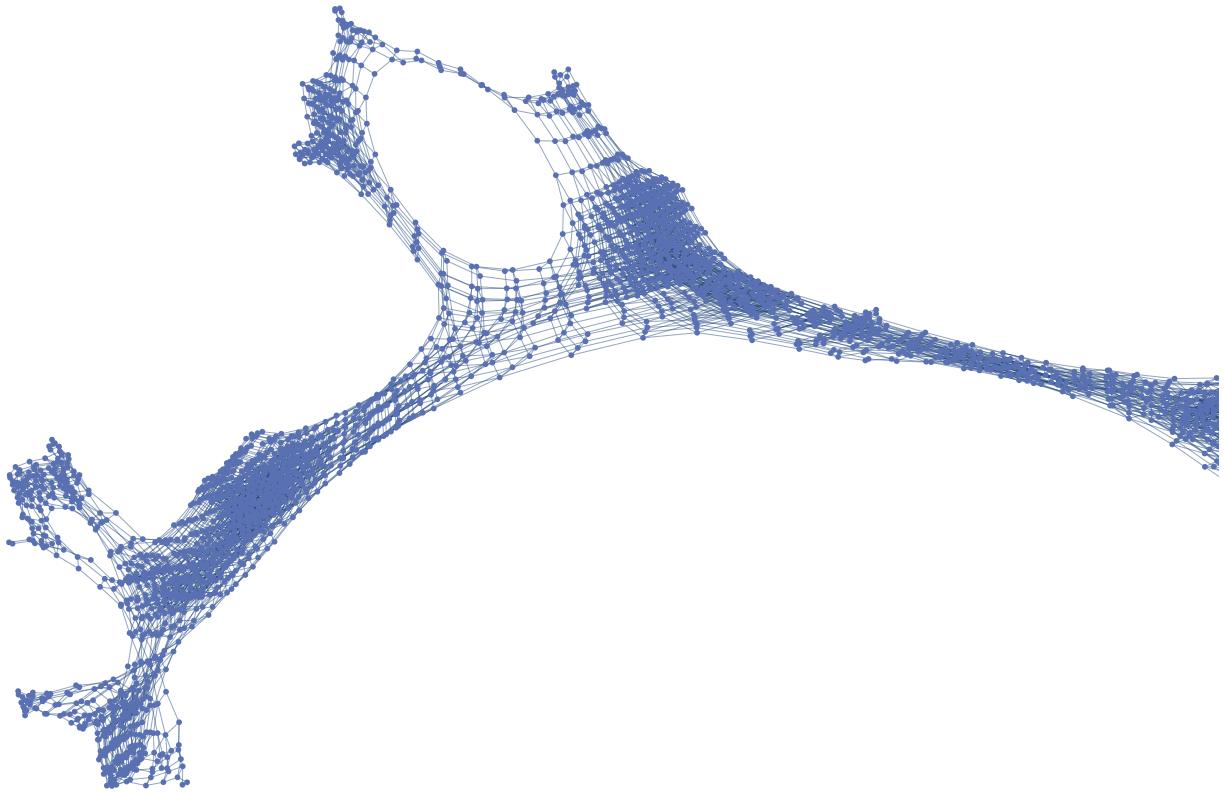
```
4846
```

```
solvedPositions = Select[VertexList@g, #[[1, "Positions"]]=={{5, 4}, {6, 4}}&];
```

```
showBoard[#, {6, 6}] & /@ RandomSample[solvedPositions, 10]
```



```
HighlightGraph[g, solvedPositions]
```



```
spf = FindShortestPath[g, cars, All]
ShortestPathFunction[
{ {<| Positions -> {{2, 4}, {3, 4}}, Directions -> {{-1, 0}, {1, 0}}|>,
  <| Positions -> {{2, 3}, {3, 3}, {4, 3}}, Directions -> {{-1, 0}, {1, 0}}|>,
  <| Positions -> {{1, 1}, {2, 1}}, Directions -> {{-1, 0}, {1, 0}}|>,
  <| Positions -> {{1, 5}, {2, 5}}, Directions -> {{-1, 0}, {1, 0}}|>,
  <| Positions -> {{1, 6}, {2, 6}}, Directions -> {{-1, 0}, {1, 0}}|>,
  <| Positions -> {{5, 6}, {6, 6}}, Directions -> {{-1, 0}, {1, 0}}|>,
  <| Positions -> {{5, 1}, {6, 1}}, Directions -> {{-1, 0}, {1, 0}}|>,
  <| Positions -> {{5, 2}, {6, 2}}, Directions -> {{-1, 0}, {1, 0}}|>,
  <| Positions -> {{1, 2}, {1, 3}, {1, 4}}, Directions -> {{0, -1}, {0, 1}}|>,
  <| Positions -> {{6, 3}, {6, 4}, {6, 5}}, Directions -> {{0, -1}, {0, 1}}|>,
  <| Positions -> {{4, 1}, {4, 2}}, Directions -> {{0, -1}, {0, 1}}|>,
  <| Positions -> {{3, 5}, {3, 6}}, Directions -> {{0, -1}, {0, 1}}|> }, All], <>>]
```

```
solvedPath = First@MinimalBy[spf /. solvedPositions, Length]
```

```
{ {<| Positions → {{2, 4}, {3, 4}}, Directions → {{-1, 0}, {1, 0}}|>,  
<| Positions → {{2, 3}, {3, 3}, {4, 3}}, Directions → {{-1, 0}, {1, 0}}|>,  
<| Positions → {{1, 1}, {2, 1}}, Directions → {{-1, 0}, {1, 0}}|>,  
<| Positions → {{1, 5}, {2, 5}}, Directions → {{-1, 0}, {1, 0}}|>,  
<| Positions → {{1, 6}, {2, 6}}, Directions → {{-1, 0}, {1, 0}}|>,  
<| Positions → {{5, 6}, {6, 6}}, Directions → {{-1, 0}, {1, 0}}|>,  
<| Positions → {{5, 1}, {6, 1}}, Directions → {{... 1 ...}, ... 1 ...}|>,  
<| Positions → {{5, 2}, {6, 2}}, Directions → {{-1, 0}, {1, 0}}|>,  
<| Positions → {{1, 2}, {1, 3}, {1, 4}}, Directions → {{0, -1}, {0, 1}}|>,  
<| Positions → {{6, 3}, {6, 4}, {6, 5}}, Directions → {{0, -1}, {0, 1}}|>,  
<| Positions → {{4, 1}, {4, 2}}, Directions → {{0, -1}, {0, 1}}|>,  
<| Positions → {{3, 5}, {3, 6}}, Directions → {{0, -1}, {0, 1}}|>},  
... 31 ..., {... 1 ...}}
```

large output

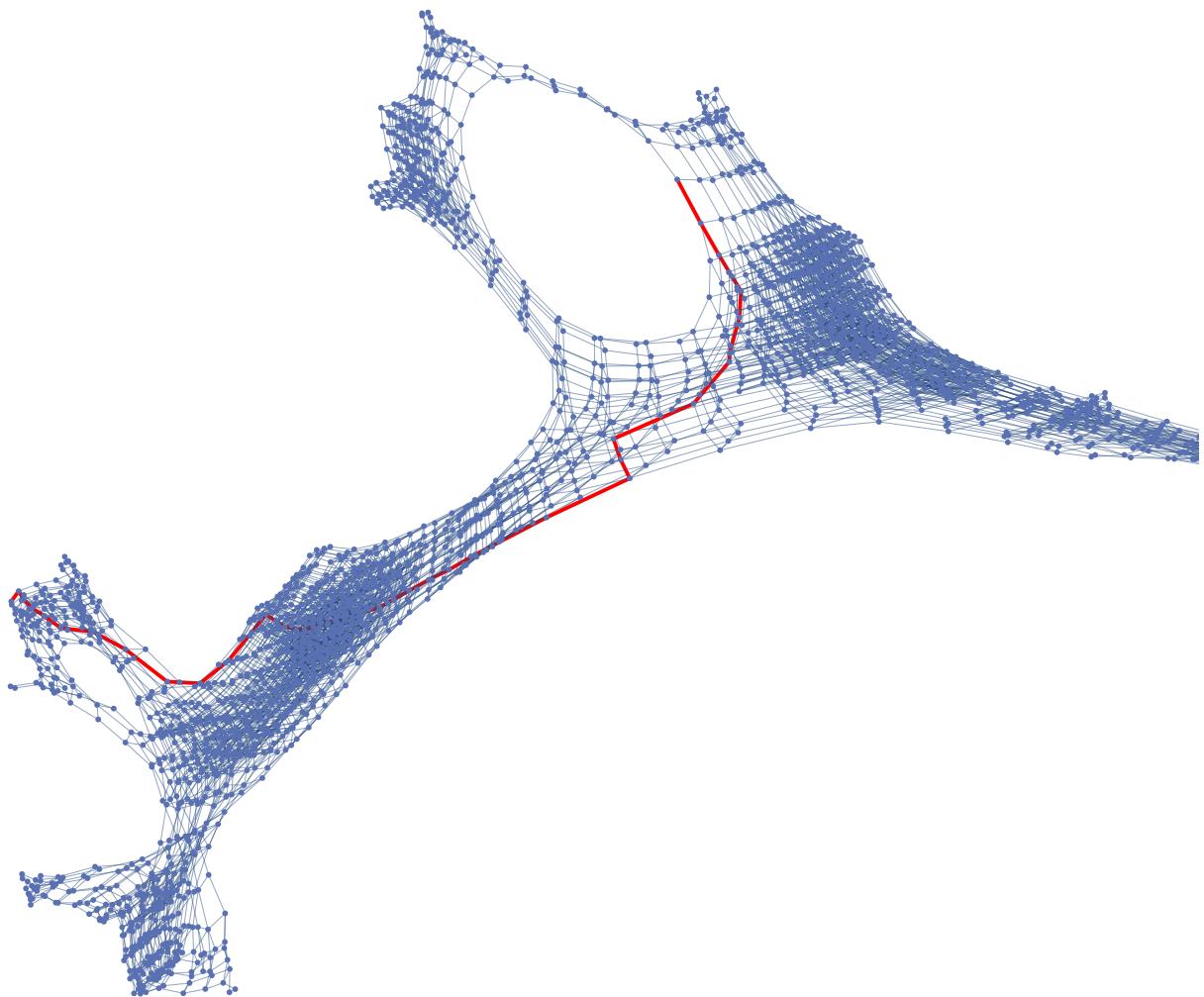
[show less](#)

[show more](#)

[show all](#)

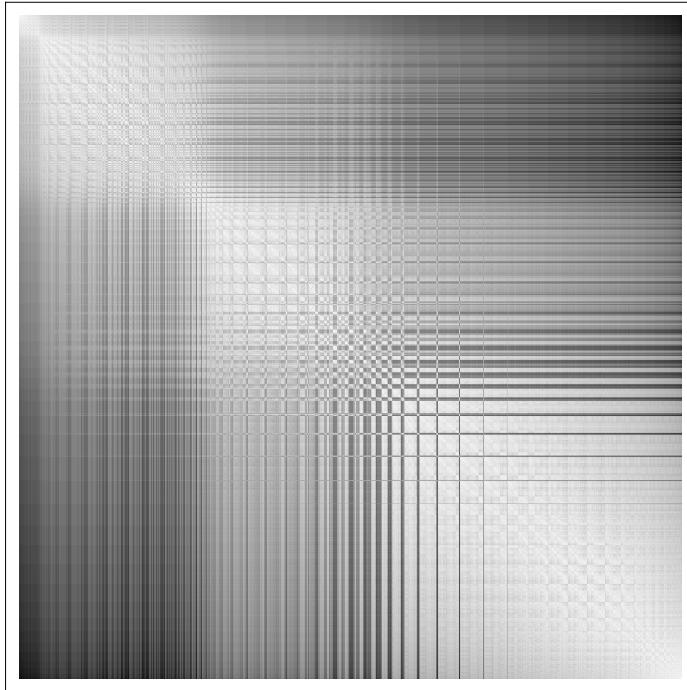
[set size limit...](#)

```
HighlightGraph[g, Style[PathGraph@solvedPath, Thick, Red, Opacity[1]]]
```

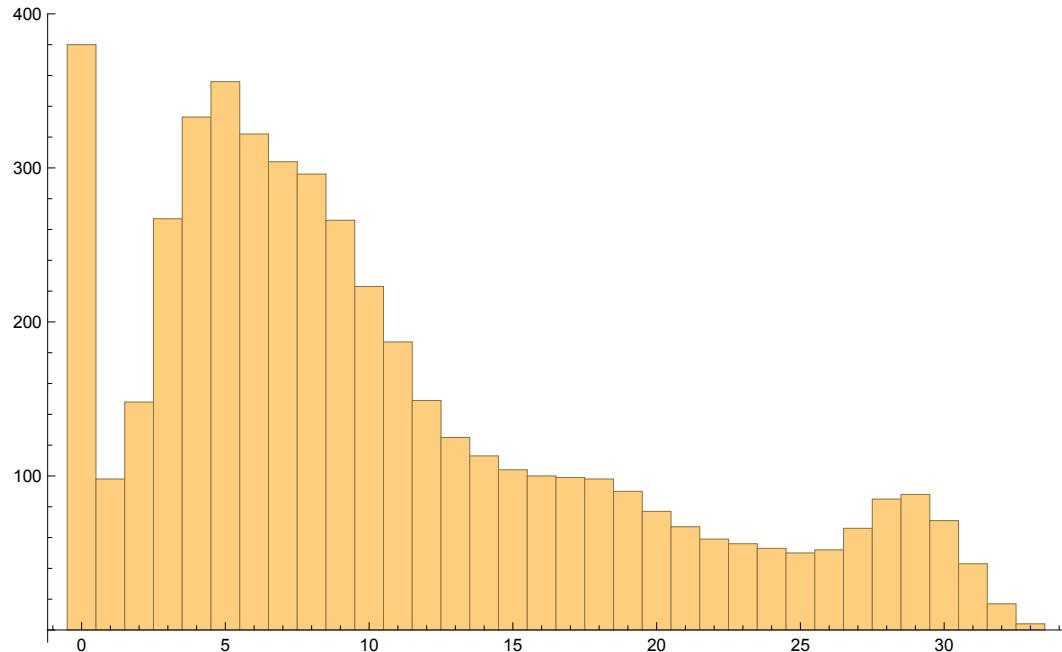


```
dm = GraphDistanceMatrix[g];
```

```
ArrayPlot@dm
```

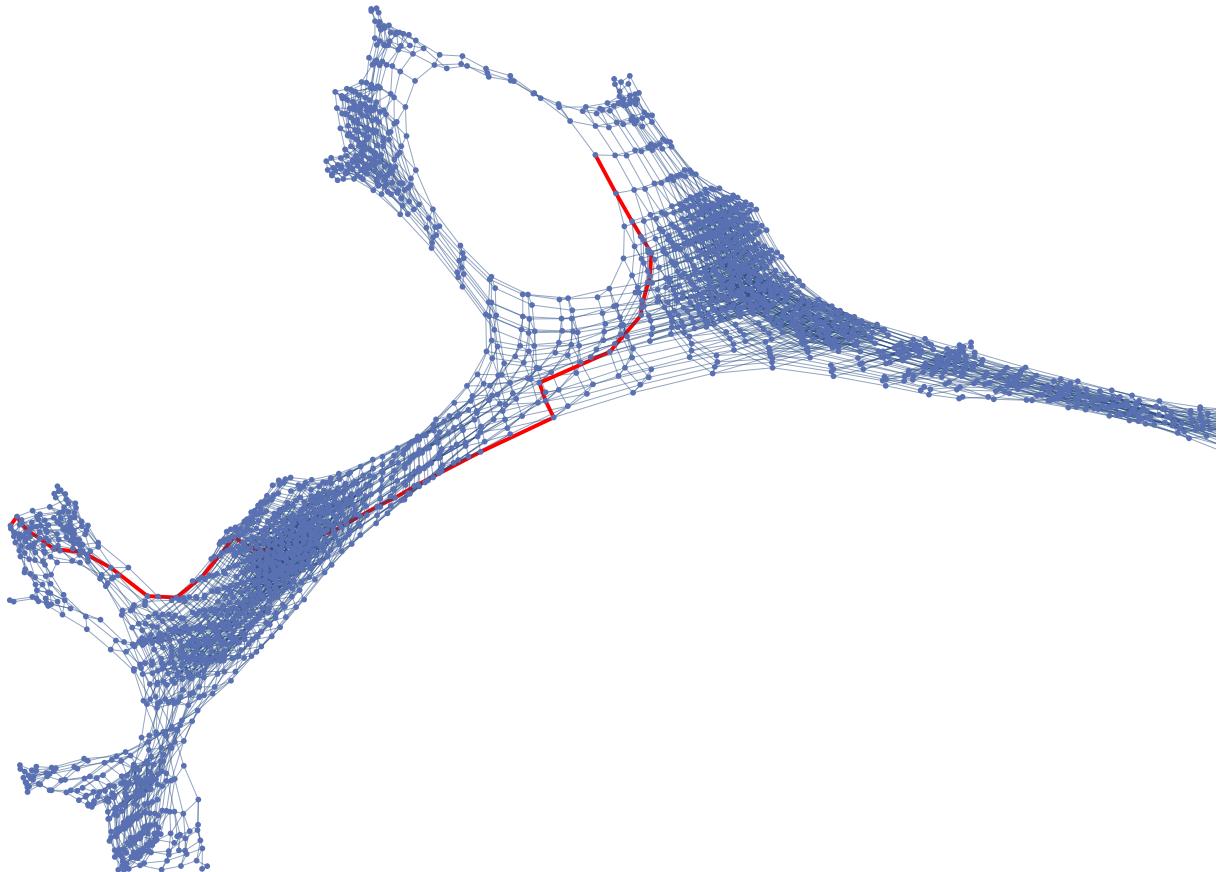


```
Histogram[Min/@  
dm[[All, Position[VertexList[g], Alternatives@@solvedPositions][[All, 1]]]], {1}]
```



```
Ordering[Min/@  
dm[[All, Position[VertexList[g], Alternatives@@solvedPositions][[All, 1]]]], 1]  
{2251}
```

```
HighlightGraph[g,
Style[PathGraph@spf[VertexList@g][[2251]], Thick, Red, Opacity[1]]]
```



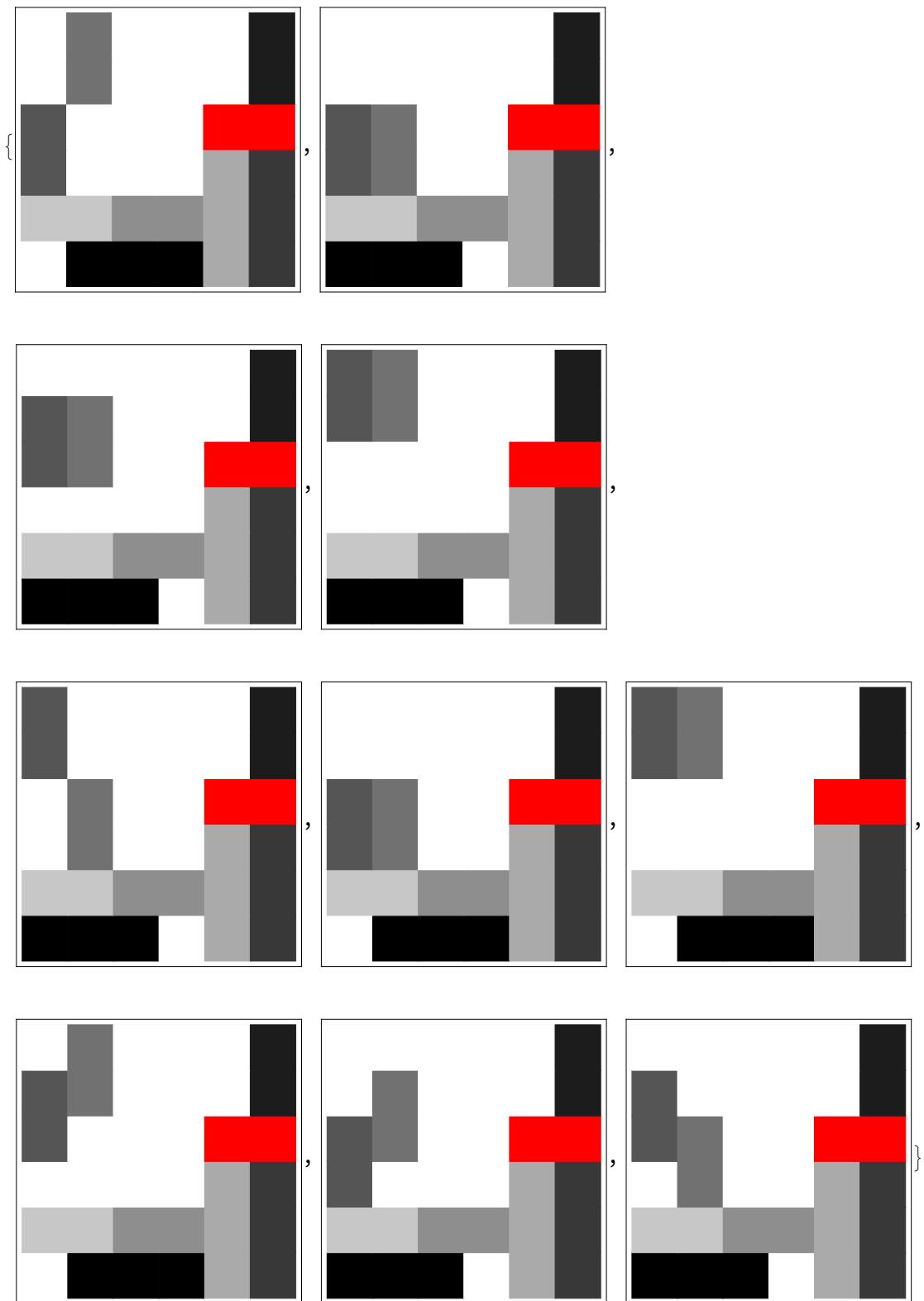
Random board

```
g = Graph@
DeleteDuplicatesBy[UndirectedEdge @@ Rest@FixedPoint[posss \[Implies] Catenate[{posss,
Catenate[(pos \[Implies] pos \[Function] # & /@ singleMoves[pos, {6, 6}]) /@ Complement[
Values@posss, Keys@posss]]}], {Missing[] \[Implies] cars}], Sort];
```

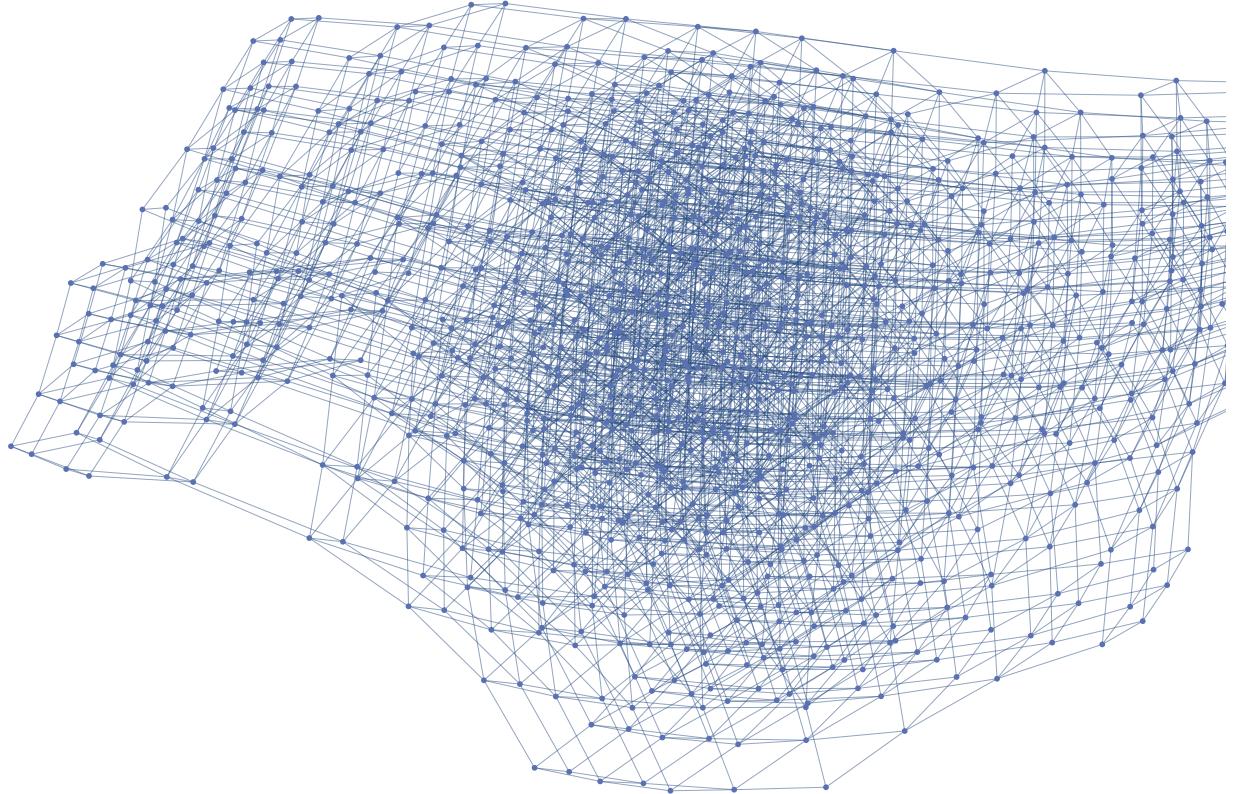
```
VertexCount@g
1230
```

```
solvedPositions = Select[VertexList@g, #[[1, "Positions"]] == {{5, 4}, {6, 4}} &];
```

```
showBoard[#, {6, 6}] & /@ RandomSample[solvedPositions, 10]
```



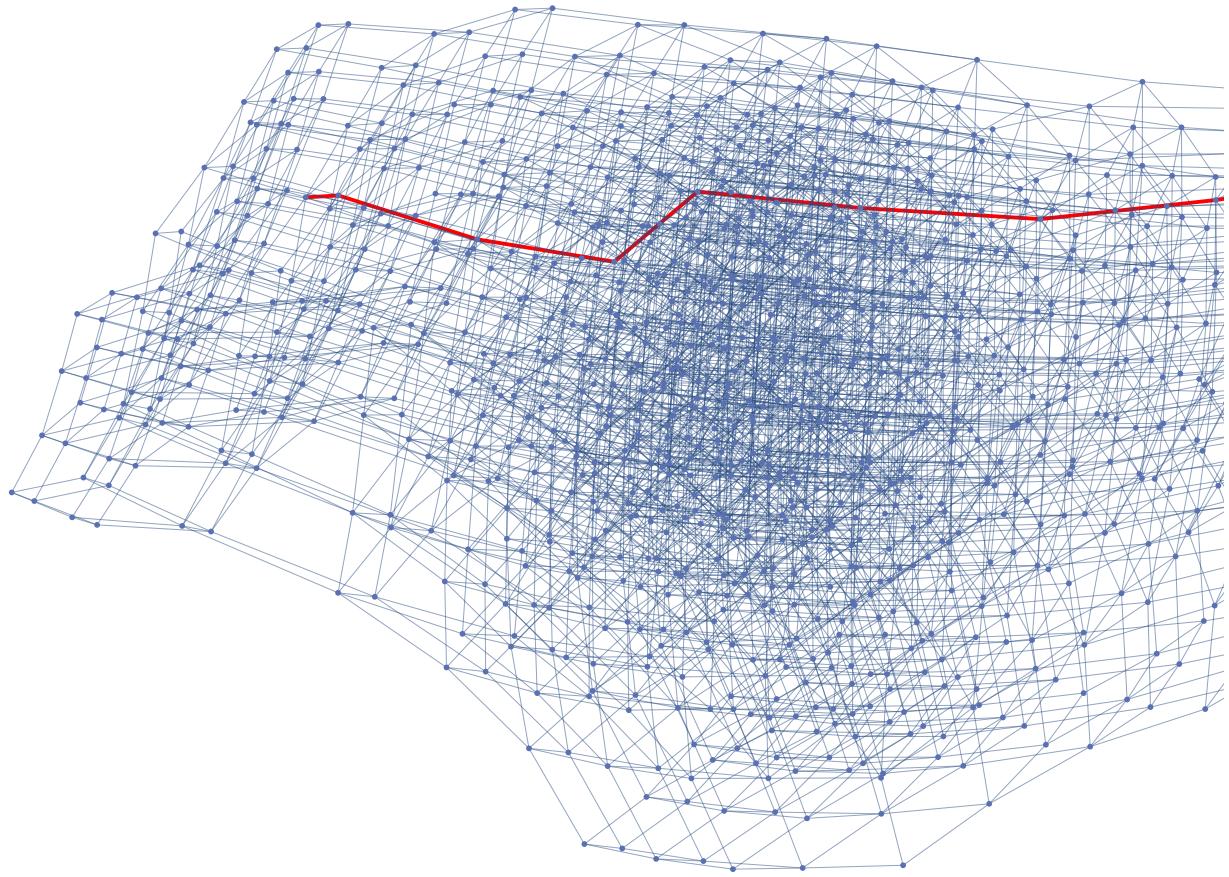
```
HighlightGraph[g, solvedPositions]
```

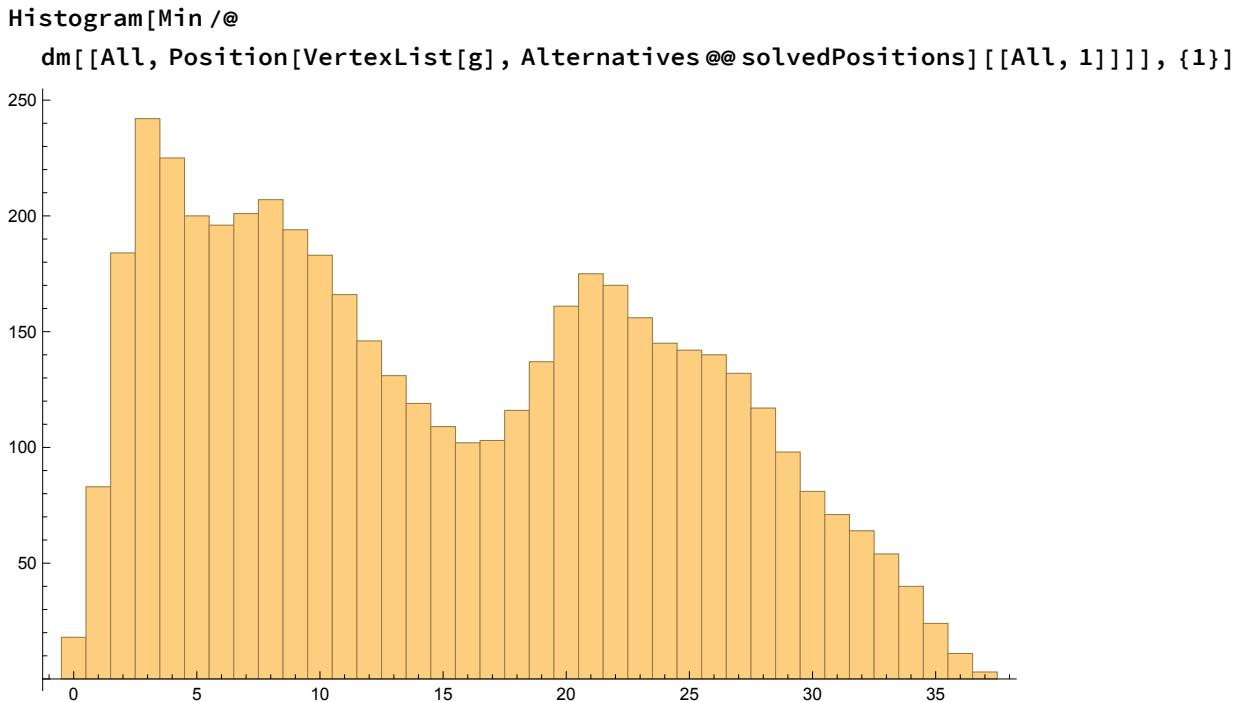


```
spf = FindShortestPath[g, cars, All]
ShortestPathFunction[
{<| Positions -> {{3, 4}, {4, 4}}, Directions -> {{-1, 0}, {1, 0}}|>,
 <| Positions -> {{2, 2}, {3, 2}}, Directions -> {{-1, 0}, {1, 0}}|>,
 <| Positions -> {{5, 4}, {5, 5}, {5, 6}}, Directions -> {{0, -1}, {0, 1}}|>,
 <| Positions -> {{4, 2}, {5, 2}}, Directions -> {{-1, 0}, {1, 0}}|>,
 <| Positions -> {{2, 3}, {2, 4}}, Directions -> {{0, -1}, {0, 1}}|>,
 <| Positions -> {{1, 2}, {1, 3}}, Directions -> {{0, -1}, {0, 1}}|>,
 <| Positions -> {{6, 2}, {6, 3}, {6, 4}}, Directions -> {{0, -1}, {0, 1}}|>,
 <| Positions -> {{6, 5}, {6, 6}}, Directions -> {{0, -1}, {0, 1}}|>,
 <| Positions -> {{3, 1}, {4, 1}, {5, 1}},
 Directions -> {{-1, 0}, {1, 0}}|> }, All], <>>]

solvedPath = First@MinimalBy[spf /. solvedPositions, Length];
```

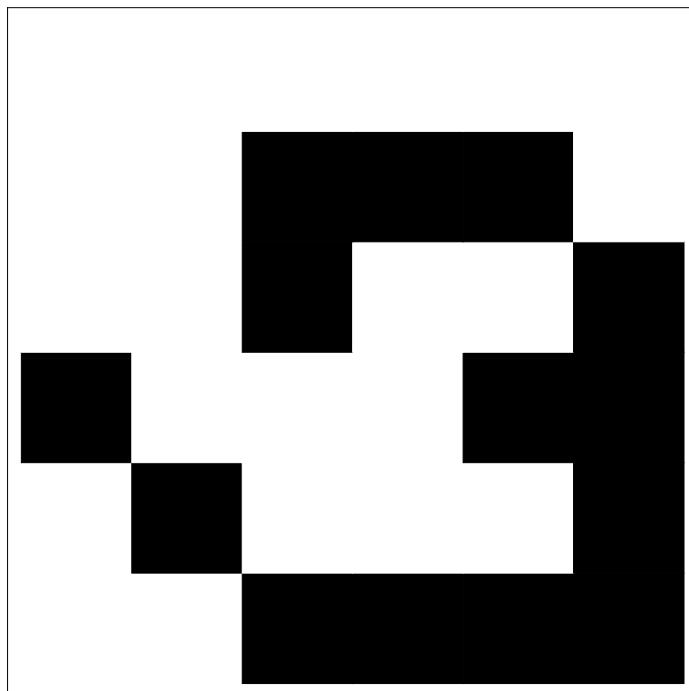
```
HighlightGraph[g, Style[PathGraph@solvedPath, Thick, Red, Opacity[1]]]
```





Enumeration of Boards

```
ArrayPlot@Partition[IntegerDigits[237384783, 2, 36], 6]
```



<http://www.ulb.ac.be/di/algo/secollet/papers/crs06.pdf>

```
Round[3.6 * 10^10]
```

```
36 000 000 000
```

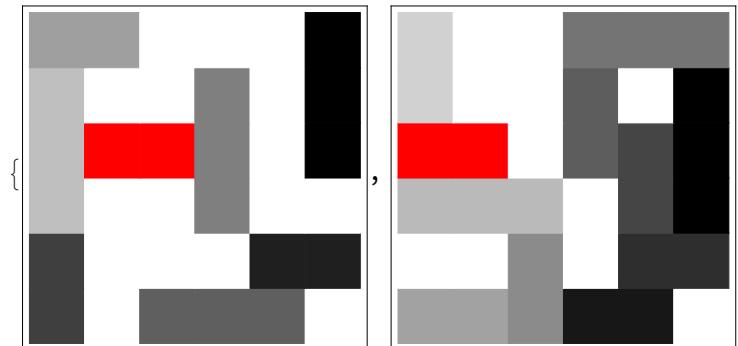
Thinkfun Boards

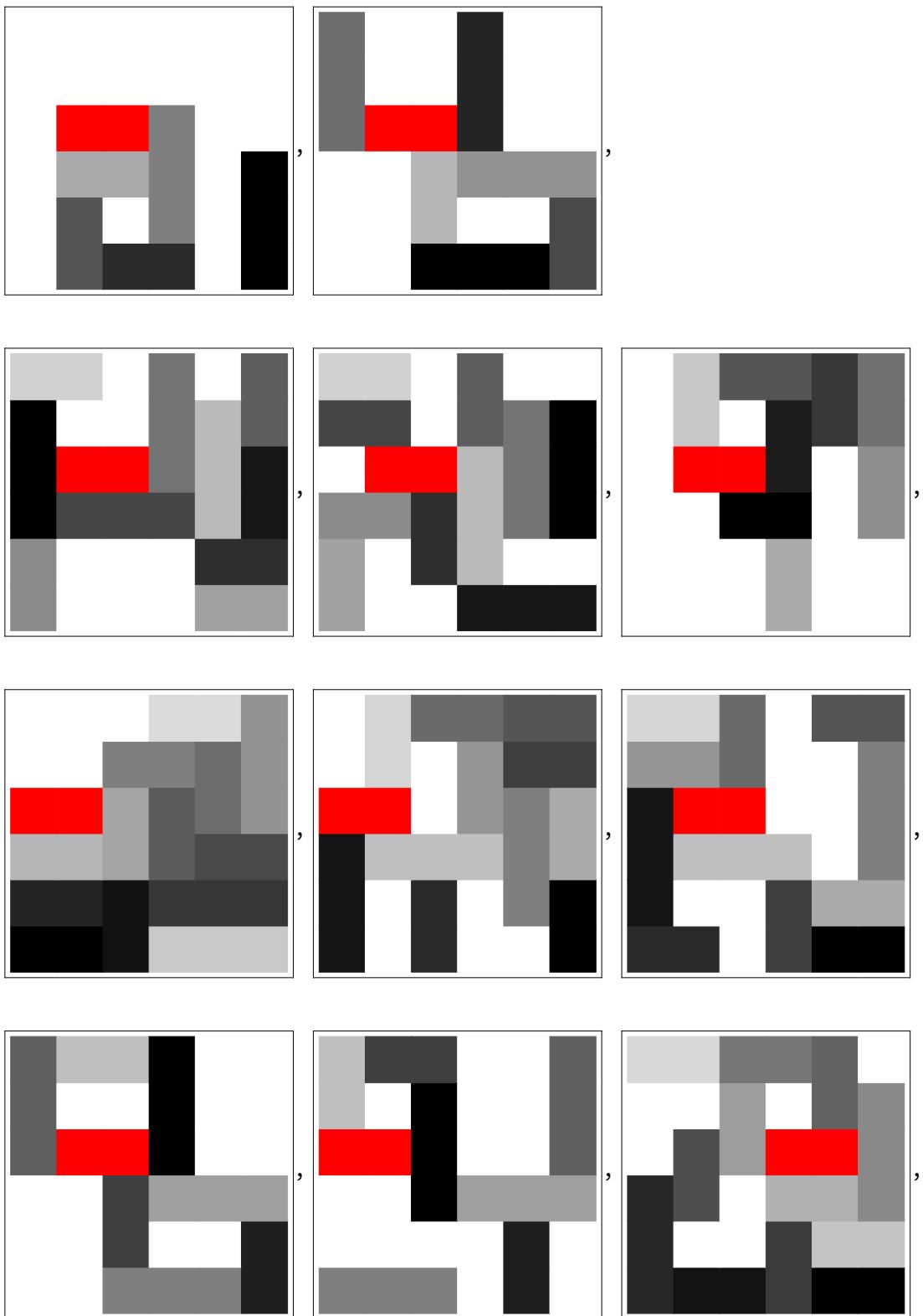
<http://www.mathsonline.org/game/jam.html?16>

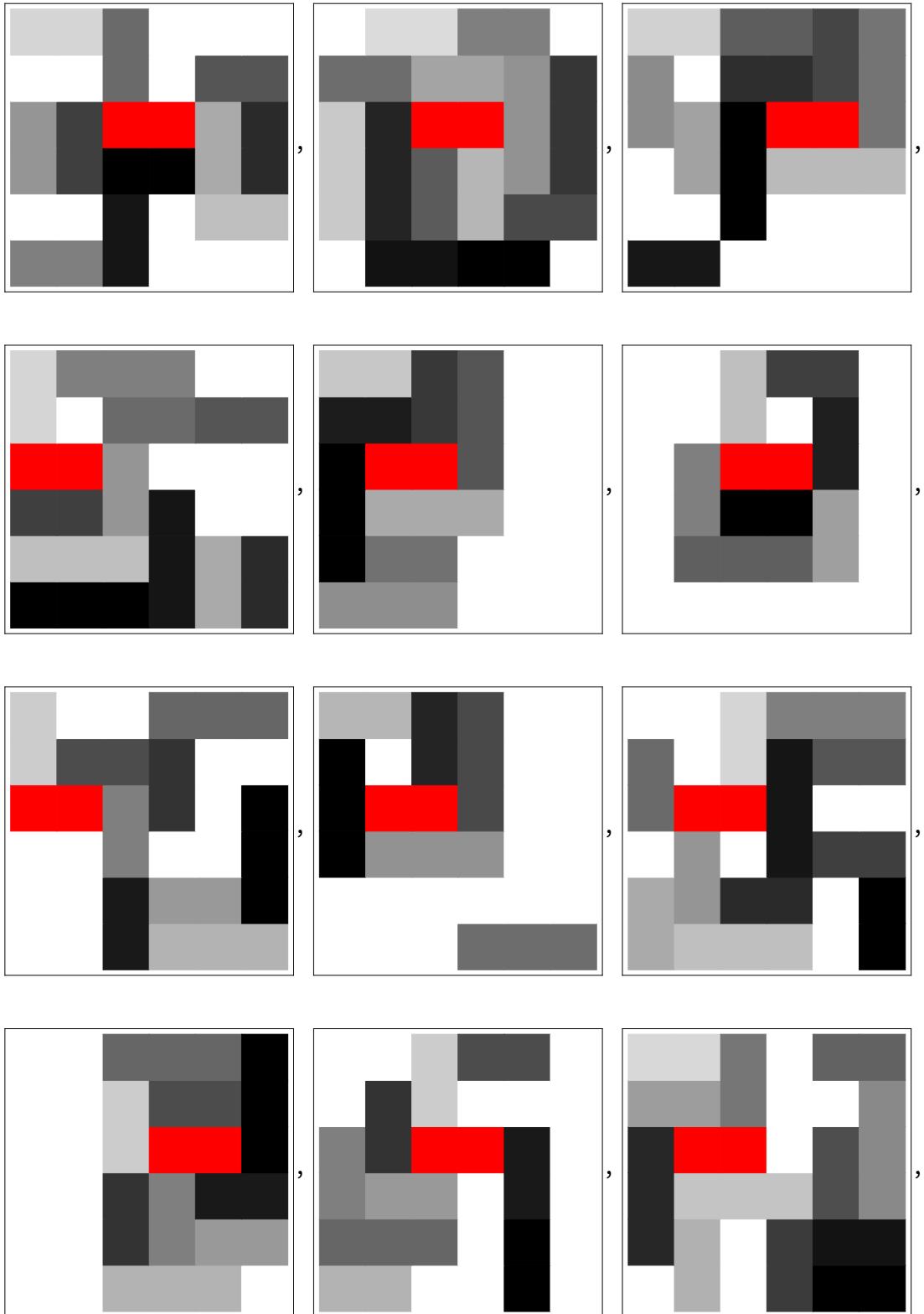
<http://www.mathsonline.org/game/jam16.js>

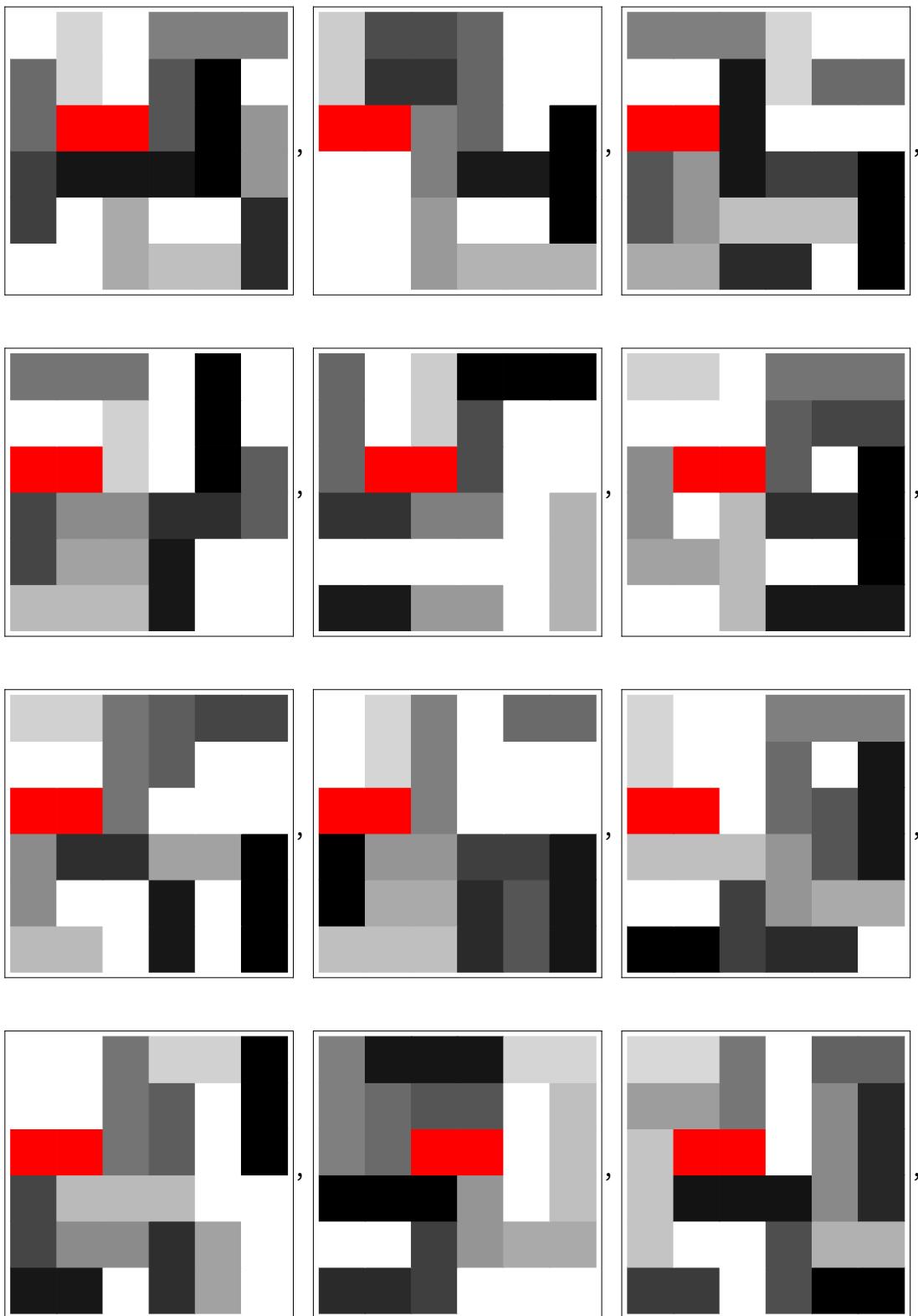
```
boards = Function[board, StringCases[
  URLRead["http://www.mathsonline.org/game/jam"<>ToString[board]<>.js", "Body"],
  Shortest["vehicle["~~__~~"] = new car(\"~~name___~\"\",(fTop+"~~top__~" *72),(fLeft+
  Block[{root, dir},
    root = {ToExpression[left]+1, 6-ToExpression[top]}];
    dir = StringTake[name,{-6,-5}];
    <|
      "Positions" → {root, root+If[dir==="EW",{1,0},{0,-1}], If[StringContainsQ[nam
      "Directions" → Replace[dir, {"EW" → {{-1,0},{1,0}}, "NS" → {{0,-1},{0,1}}}]
    |>
  ]
  ]]
]] /@ Range[40];

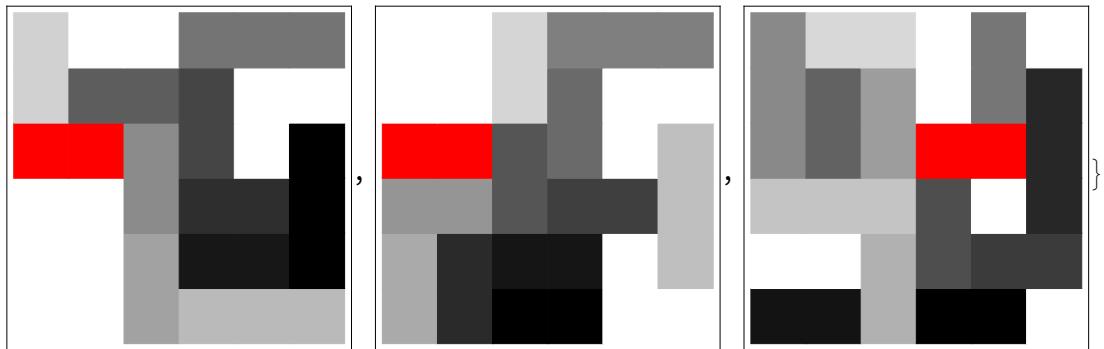
boards = Get["~/Desktop/boards.wl"];
showBoard[#, {6, 6}] & /@ boards
```











```

boards[[1]] = 
{<| Positions -> {{2, 4}, {3, 4}}, Directions -> {{-1, 0}, {1, 0}}|>,
 <| Positions -> {{1, 5}, {1, 4}, {1, 3}}, Directions -> {{0, -1}, {0, 1}}|>,
 <| Positions -> {{1, 6}, {2, 6}}, Directions -> {{-1, 0}, {1, 0}}|>,
 <| Positions -> {{4, 5}, {4, 4}, {4, 3}}, Directions -> {{0, -1}, {0, 1}}|>,
 <| Positions -> {{3, 1}, {4, 1}, {5, 1}}, Directions -> {{-1, 0}, {1, 0}}|>,
 <| Positions -> {{1, 2}, {1, 1}}, Directions -> {{0, -1}, {0, 1}}|>,
 <| Positions -> {{5, 2}, {6, 2}}, Directions -> {{-1, 0}, {1, 0}}|>,
 <| Positions -> {{6, 6}, {6, 5}, {6, 4}}, Directions -> {{0, -1}, {0, 1}}|>}

moveGraph[board_, boardSize_: {6, 6}] :=
Graph@DeleteDuplicatesBy[UndirectedEdge @@ Rest@FixedPoint[poss &gt;
Catenate[{poss, Catenate[(pos &gt; pos -> # & /@ singleMoves[pos, boardSize]) /@
Complement[Values@poss, Keys@poss]]}], {Missing[] &gt; board}], Sort]

ListPlot[Length /@ boards]

```

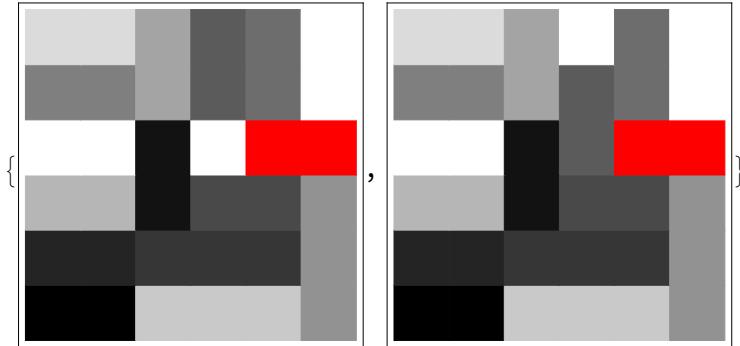
```

moveGraph[boards[[2]]]
$Aborted

solvedPositions[g_Graph] :=
Select[VertexList[g], #[[1, "Positions"]] === {{5, 4}, {6, 4}} &]

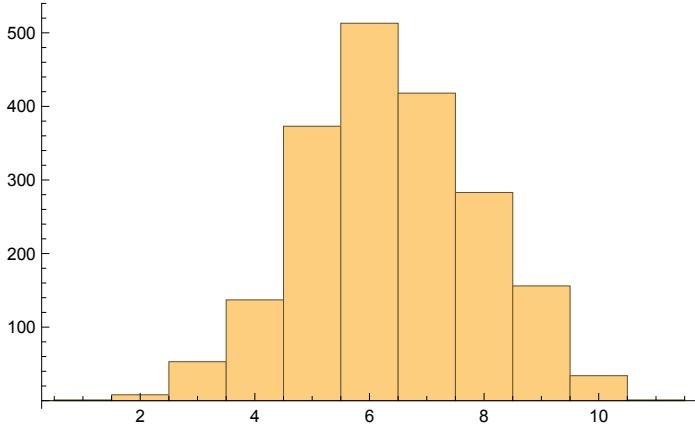
```

```
showBoard[#, {6, 6}] & /@ solvedPositions[moveGraph[boards[[8]]]]
```



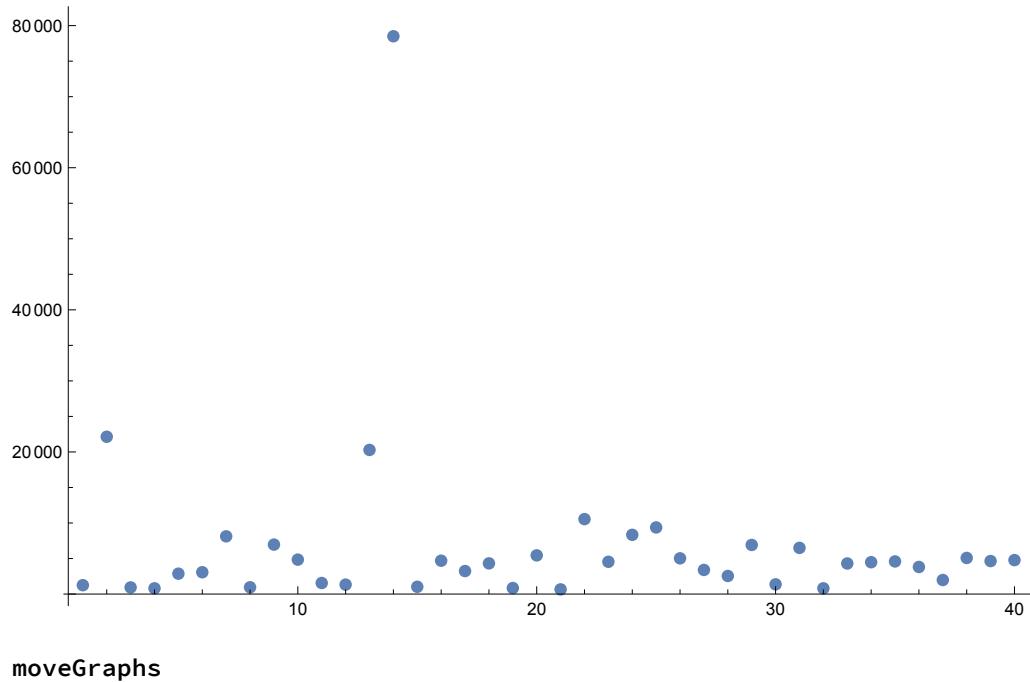
```
moveGraphs = moveGraph /@ boards;
(*moveGraphs>>"~/Desktop/moveGraphs.wl";*)
(*DumpSave["~/Desktop/moveGraphs.mx",moveGraphs];*)
Get["~/Desktop/moveGraphs.mx"]
FindDistribution@VertexDegree[moveGraphs[[-1]]]
BinomialDistribution[11, 0.562011]
```

```
Histogram[VertexDegree[moveGraphs[[-4]]]]
```

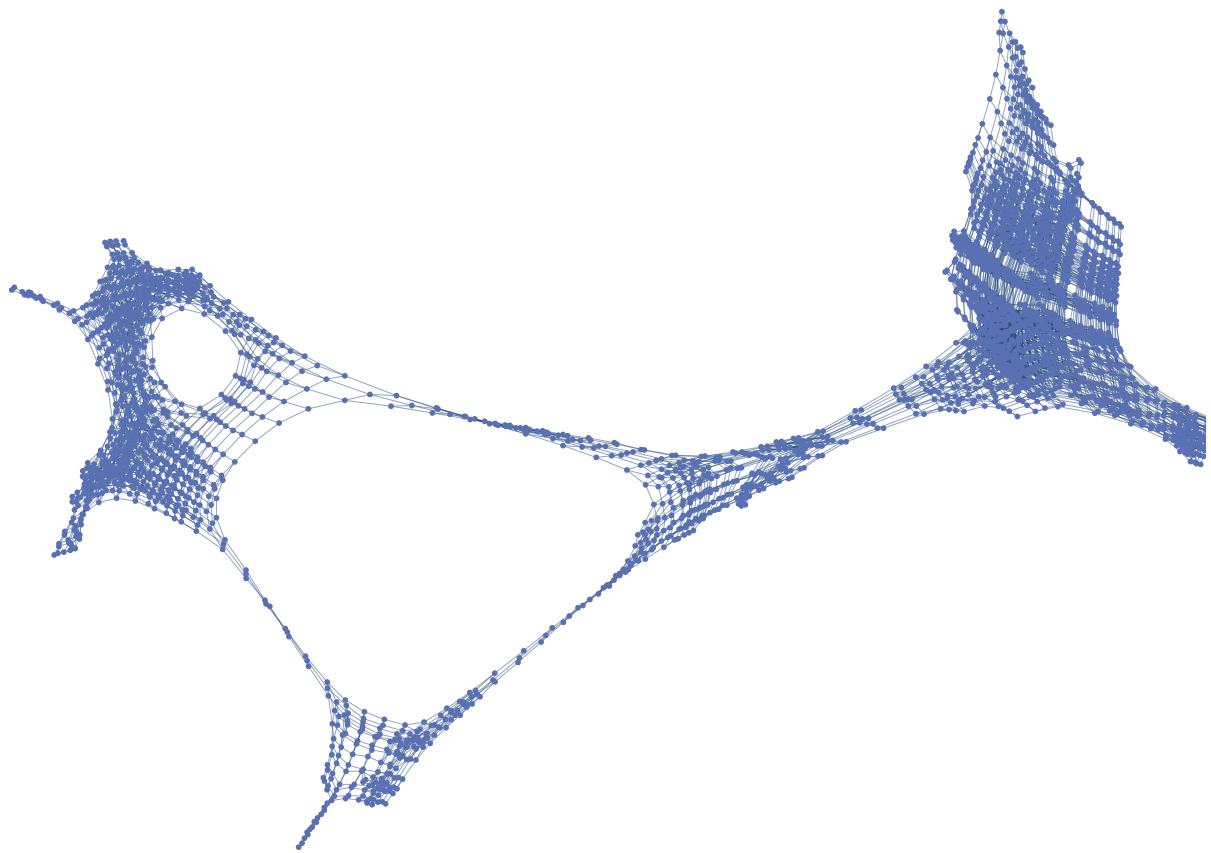


```
moveGraph[boards[[40]]];
```

```
ListPlot[VertexCount /@ moveGraphs, PlotRange -> All]
```

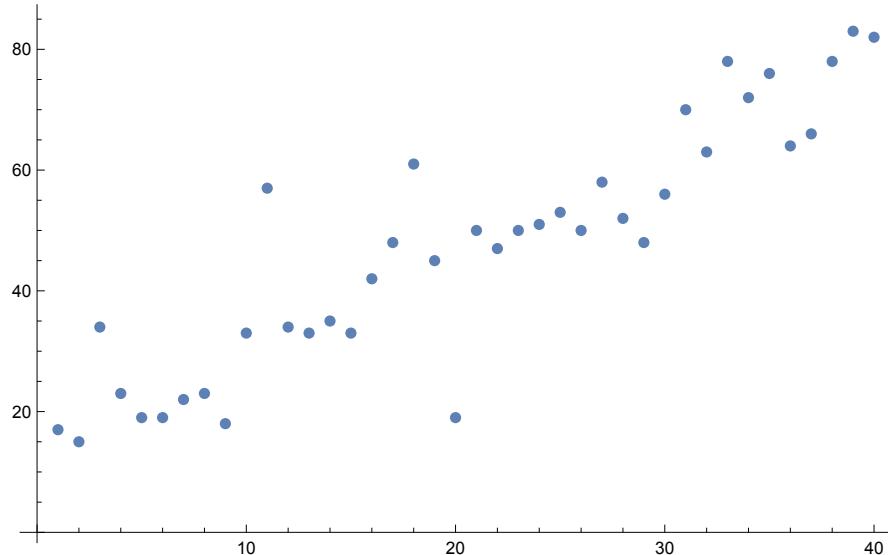


```
HighlightGraph[moveGraphs[[-1]],  
Style[#, Red, Large] & /@ solvedPositions[moveGraphs[[-1]]]]
```

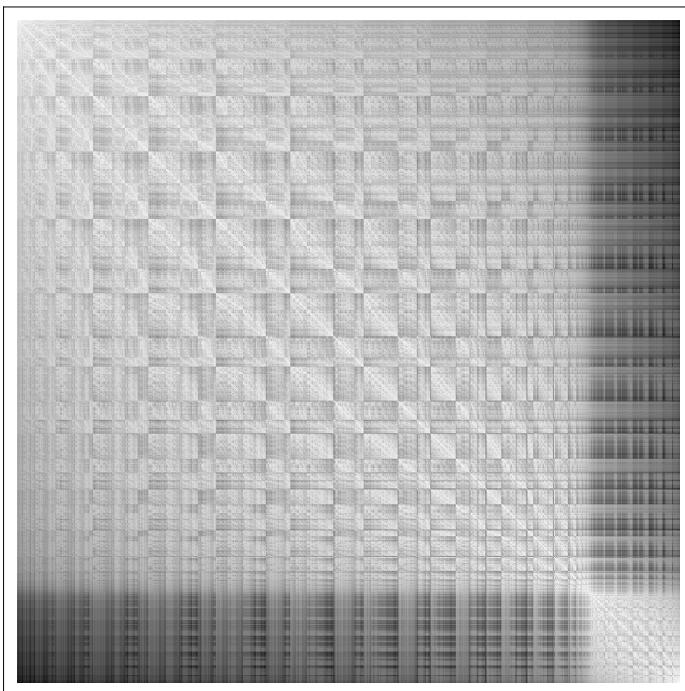


```
solutionPath[g_, init_] := Block[{spf}, spf = FindShortestPath[g, init, All];  
First@MinimalBy[spf /. solvedPositions[g], Length]]
```

```
ListPlot@MapThread[Length@solutionPath[#1, #2] &, {moveGraphs, boards}]
```



```
ArrayPlot@GraphDistanceMatrix[moveGraphs[[1]]]
```

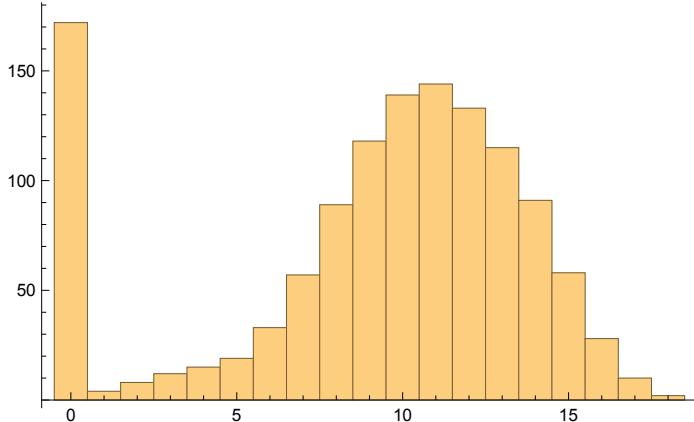


```
solutionDistances[g_] := Length@solutionPath[g, #] & /@ VertexList[g]
solutionDistances[g_] := Min /@ GraphDistanceMatrix[g][[All,
Position[VertexList[g], Alternatives @@ solvedPositions[g]][[All, 1]]]]]

solutionDistances[g_] :=
GraphDistance[IndexGraph[g], First@FirstPosition[VertexList[g], #]] & /@
solvedPositions[g]
```

```
solutionDistances[moveGraphs[[1]]]
```

```
Histogram[Min/@Transpose[%42]]
```



```
solvedPositions[moveGraphs[[14]]] // Length
```

14 647

Now

```
RepeatedTiming@GraphDistance[IndexGraph[moveGraphs[[2]]],  
First@FirstPosition[VertexList[moveGraphs[[2]]],  
solvedPositions[moveGraphs[[2]]][[1]]], First@FirstPosition[  
VertexList[moveGraphs[[2]]], solvedPositions[moveGraphs[[2]]][[-1]]]]  
{2.571, 48}
```

```
Length[solvedPositions[moveGraphs[[2]]]]
```

1084

$2.570829999999999486`4. * 1084 / 60 / 60$

0.7741

$3.7904714999999997715`3. * 14\ 647 / 60 / 60$

15.4

Now

```
RepeatedTiming@GraphDistanceMatrix[moveGraphs[[1]]]
```

{0.474, {...1...}}

large output

show less

show more

show all

set size limit...

```
Length@solvedPositions[moveGraphs[[1]]]
```

172

```
AbsoluteTiming@solutionDistances[moveGraphs[[1]]]
```

```
{30.479, { ... 1 ... }}
```

large output

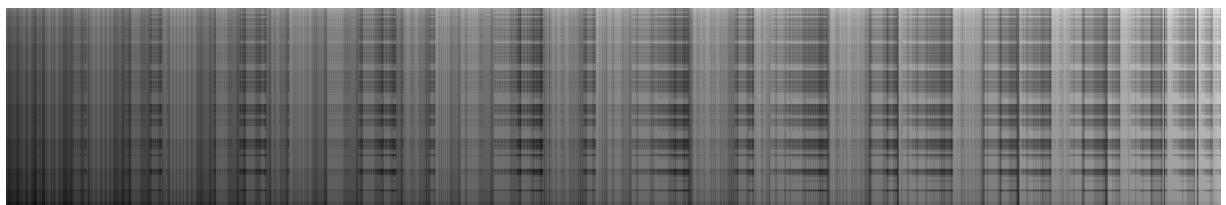
show less

show more

show all

set size limit...

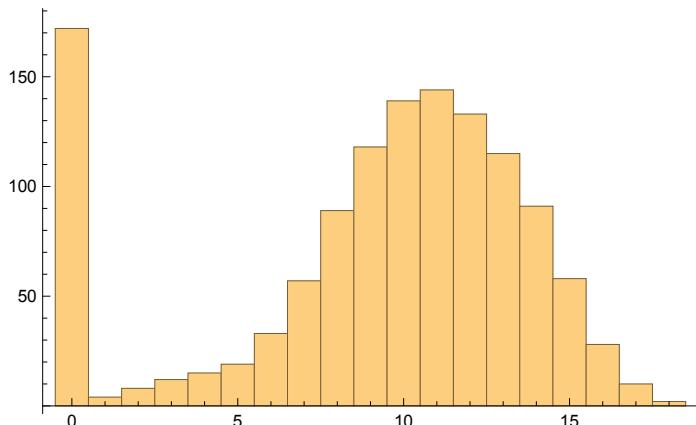
```
solutionDistances[moveGraphs[[1]]] // ArrayPlot
```



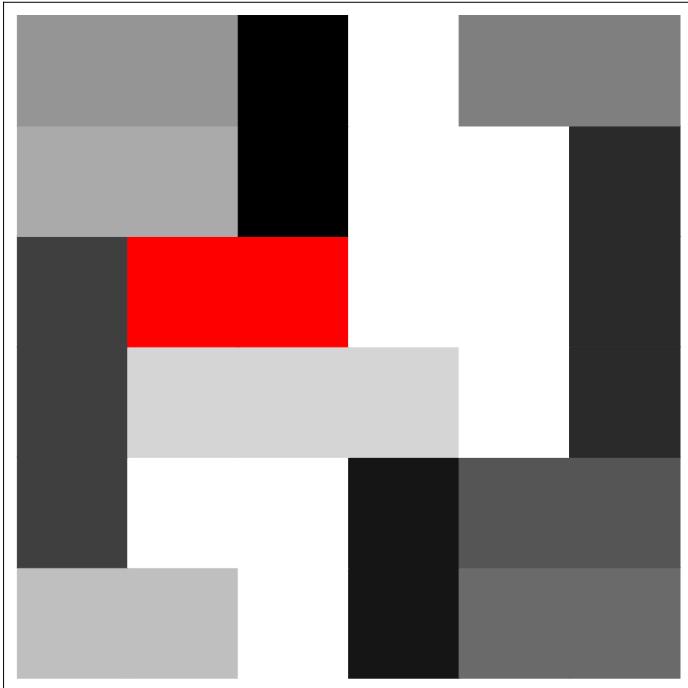
```
Histogram[solutionDistances[moveGraphs[[1]]]]
```

```
Histogram[Min/@
```

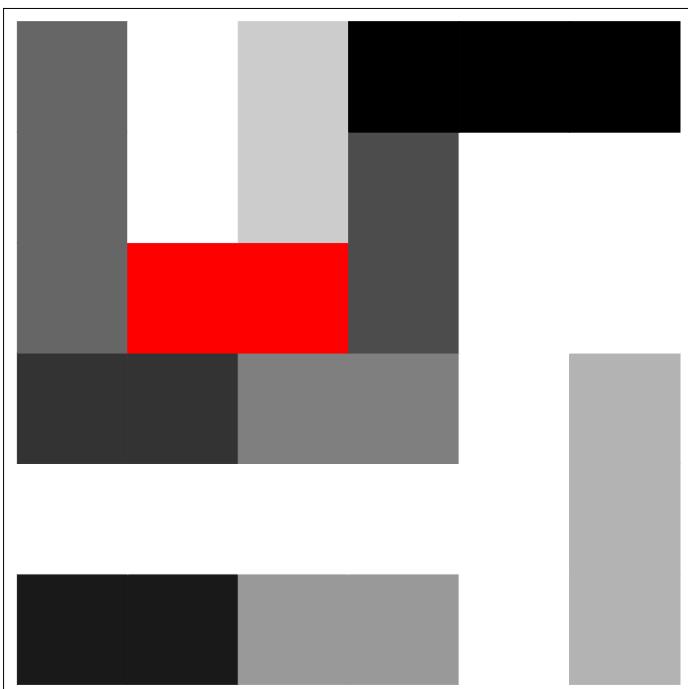
```
GraphDistanceMatrix[moveGraphs[[1]]][[All, Position[VertexList[moveGraphs[[1]]],  
Alternatives @@ solvedPositions[moveGraphs[[1]]]][[All, 1]]]]]
```



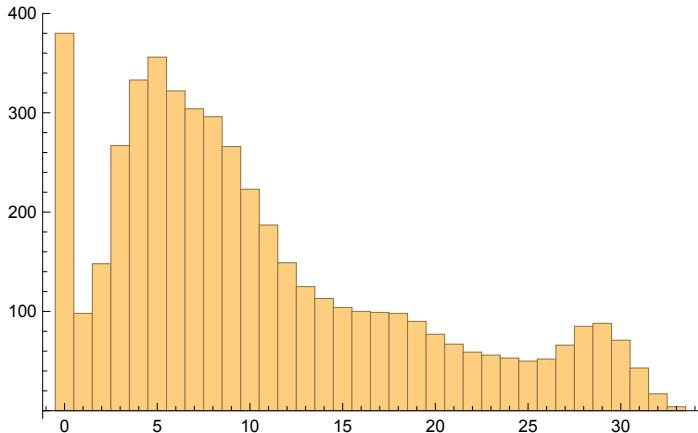
```
showBoard[cars, {6, 6}]
```



```
showBoard[boards[[30]], {6, 6}]
```



```
Histogram[solutionDistances[moveGraphs[[10]]], {1}]
```



```
means = Mean@*solutionDistances /@ moveGraphs;
```

```
Total[VertexCount /@ moveGraphs]
```

```
266 842
```

```
Histogram[Min @
dm[[All, Position[VertexList[g], Alternatives @@ solvedPositions][[All, 1]]]], {1}]
```

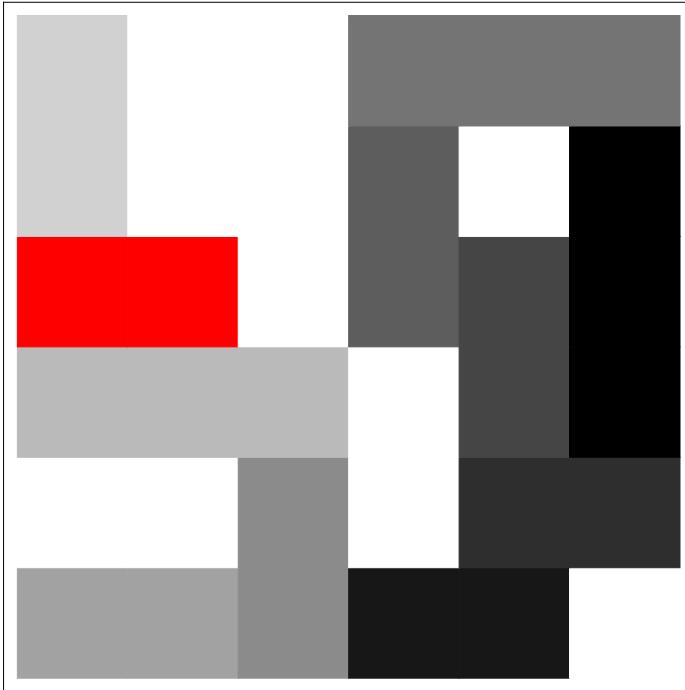
```
VertexCount@moveGraphs[[14]]
```

```
78 501
```

```
78 501^2 * 16
```

```
98 598 512 016
```

```
showBoard[boards[[2]], {6, 6}]
```



```
Histogram[solutionDistances[moveGraphs[[2]]], {1}]
```

\$Aborted

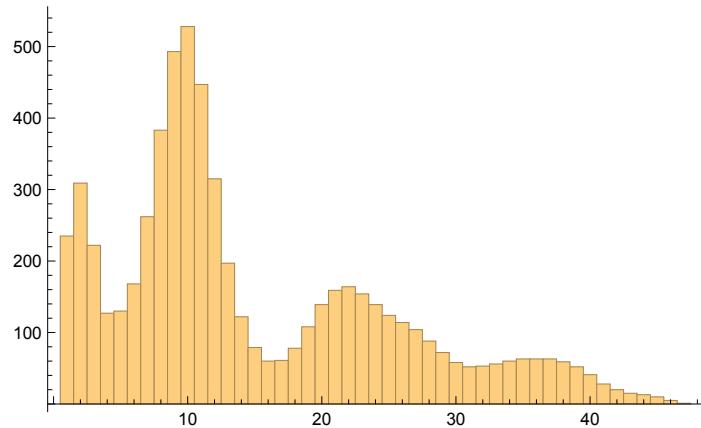
```
FileNames["~/Desktop/rushHour/*.wl"]
```

```
{~/Desktop/rushHour/10.wl, ~/Desktop/rushHour/11.wl,
~/Desktop/rushHour/12.wl, ~/Desktop/rushHour/13.wl,
~/Desktop/rushHour/14.wl, ~/Desktop/rushHour/15.wl, ~/Desktop/rushHour/16.wl,
~/Desktop/rushHour/17.wl, ~/Desktop/rushHour/18.wl, ~/Desktop/rushHour/19.wl,
~/Desktop/rushHour/1.wl, ~/Desktop/rushHour/20.wl, ~/Desktop/rushHour/21.wl,
~/Desktop/rushHour/22.wl, ~/Desktop/rushHour/23.wl, ~/Desktop/rushHour/24.wl,
~/Desktop/rushHour/25.wl, ~/Desktop/rushHour/26.wl, ~/Desktop/rushHour/27.wl,
~/Desktop/rushHour/28.wl, ~/Desktop/rushHour/29.wl, ~/Desktop/rushHour/2.wl,
~/Desktop/rushHour/30.wl, ~/Desktop/rushHour/31.wl, ~/Desktop/rushHour/32.wl,
~/Desktop/rushHour/33.wl, ~/Desktop/rushHour/34.wl, ~/Desktop/rushHour/35.wl,
~/Desktop/rushHour/36.wl, ~/Desktop/rushHour/37.wl, ~/Desktop/rushHour/38.wl,
~/Desktop/rushHour/39.wl, ~/Desktop/rushHour/3.wl, ~/Desktop/rushHour/40.wl,
~/Desktop/rushHour/4.wl, ~/Desktop/rushHour/5.wl, ~/Desktop/rushHour/6.wl,
~/Desktop/rushHour/7.wl, ~/Desktop/rushHour/8.wl, ~/Desktop/rushHour/9.wl}
```

```
solutionDistances =
```

```
Get[StringTemplate["~/Desktop/rushHour/``.wl"] [#] & /@ Range[40];
```

```
Histogram[DeleteCases[solutionDistances[[22]], 0], {1}]
```



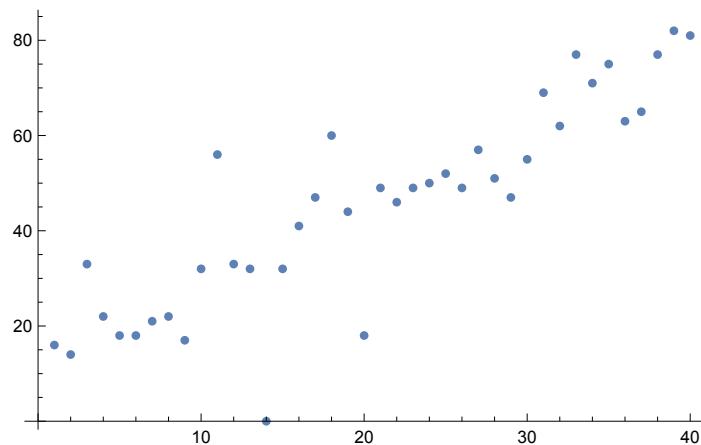
```
First@FirstPosition[VertexList[moveGraphs[[2]]], boards[[2]]]
```

```
1
```

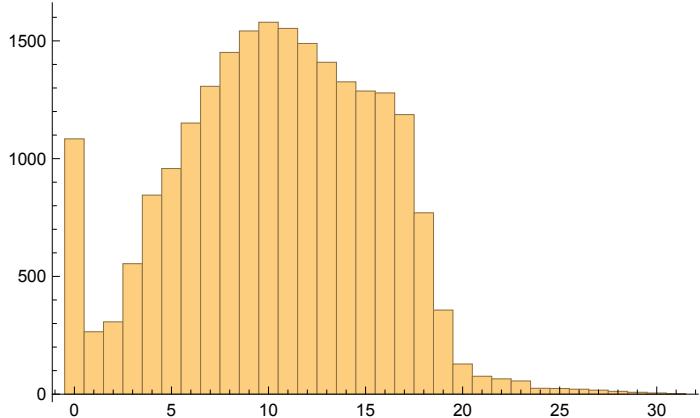
```
solutionDistances[[2, 1]]
```

```
14
```

```
ListPlot@solutionDistances[[All, 1]]
```



```
Histogram[solutionDistances[[2]]]
```



```
solutionDistances[[All, 1]]
```

```
{16, 14, 33, 22, 18, 18, 21, 22, 17, 32, 56, 33, 32, 0, 32, 41, 47, 60, 44, 18, 49, 46, 49, 50, 52, 49, 57, 51, 47, 55, 69, 62, 77, 71, 75, 63, 65, 77, 82, 81}
```

```
path = solutionPath[moveGraphs[[2]], boards[[2]]];
```

```
HighlightGraph[moveGraphs[[2]], PathGraph[path]]
```

```
HighlightGraph[moveGraphs[[2]],  
Style[PathGraph[path], Opacity[1], Thickness[0.05], Red]]
```