

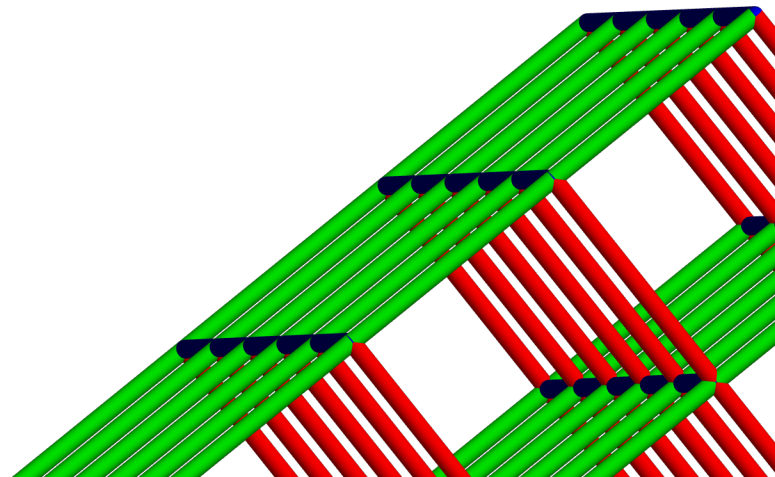
```
In[ ]:= SetDirectory["C:\\drorbn\\AcademicPensieve\\Talks\\Fields-2011"]
```

```
Out[ ]:= C:\drorbn\AcademicPensieve\Talks\Fields-2011
```

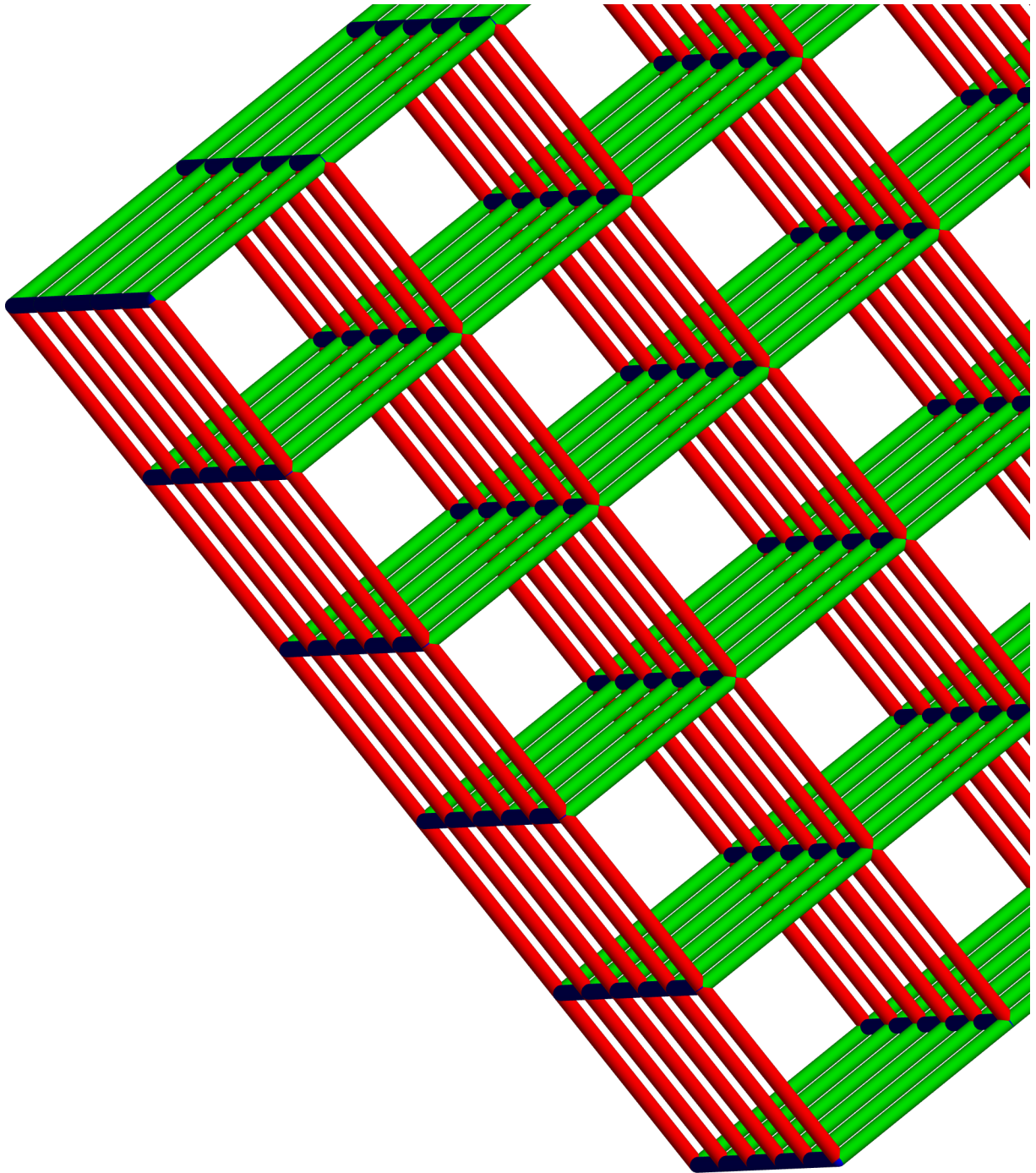
MakeImage

```
In[ ]:= Options[MakeImage] = {ImageSize -> {2400}, ThumbnailSize -> {512}};
MakeImage[g_, opts__Rule] := Module[
  {isize = ImageSize /. {opts} /. Options[MakeImage]},
  ImageResize[
    ImageCrop[Rasterize[g, RasterSize -> isize, ImageSize -> isize]],
    isize]
];
MakeImage[name_String, g_, opts__Rule] := Module[{s,
  isize = ImageSize /. {opts} /. Options[MakeImage],
  tsize = ThumbnailSize /. {opts} /. Options[MakeImage]
},
  Export[name <> ".png",
    s = ImageResize[
      ImageCrop[Rasterize[$Image[name] = g, RasterSize -> isize, ImageSize -> isize]],
      isize]
  ];
  ImageResize[s, tsize]
];
```

```
In[ ]:= n = 5;
Graphics3D[{
  Red, Table[Line[{{0, i, j}, {n, i, j}}], {i, 0, n}, {j, 0, n}],
  Green, Table[Line[{{i, 0, j}, {i, n, j}}], {i, 0, n}, {j, 0, n}],
  Blue, Table[Line[{{i, j, 0}, {i, j, n}}], {i, 0, n}, {j, 0, n}]
} /. Line -> Tube, ViewPoint -> {0, 0, ∞}, Boxed -> False]
```

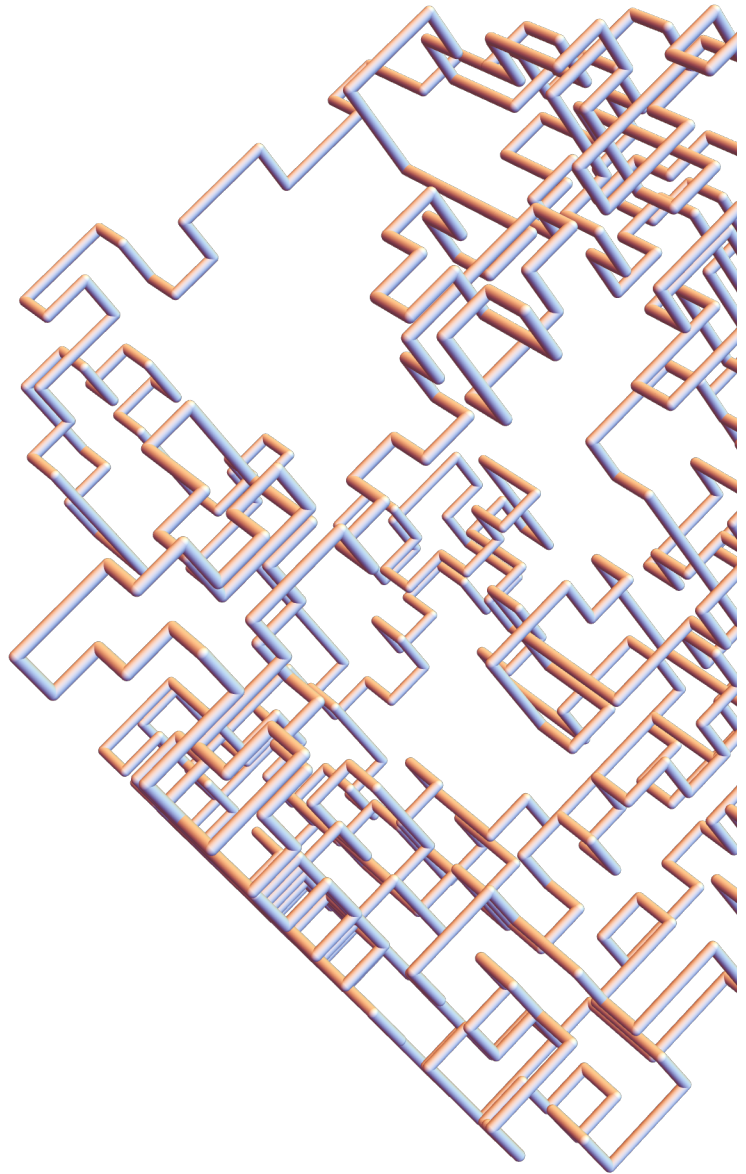


Out[]:=



```
In[*]:= n = 16; start = {1, 1, 1};
moves = {{1, 0, 0}, {-1, 0, 0}, {0, 1, 0}, {0, -1, 0}, {0, 0, 1}, {0, 0, -1}};
visited = Table[False, {n}, {n}, {n}];
path = {};
visited[[Sequence@@ (at = start)]] = True;
dirs = Range[6];
While[Length[dirs] > 0,
  d = RandomChoice[dirs];
  If[(And@@ ((1 ≤ # ≤ n) & /@ (next = at + moves[[d]]))) ∧ ! Extract[visited, next],
    AppendTo[path, Tube[{at, at = next}]]];
  visited[[Sequence@@ at]] = True;
  dirs = Range[6],
  (*else*) dirs = DeleteCases[dirs, d]
];
Graphics3D[path, Boxed → False]
```

Out[]=

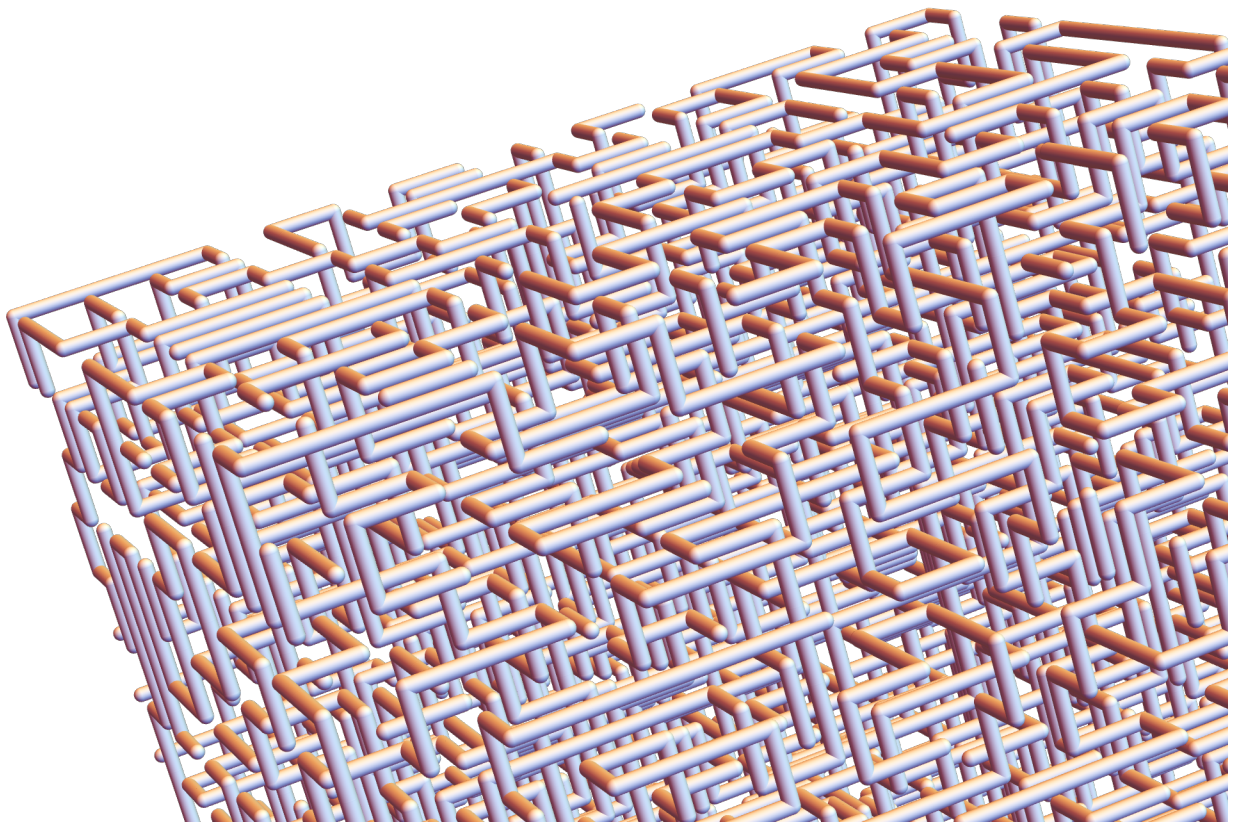



```

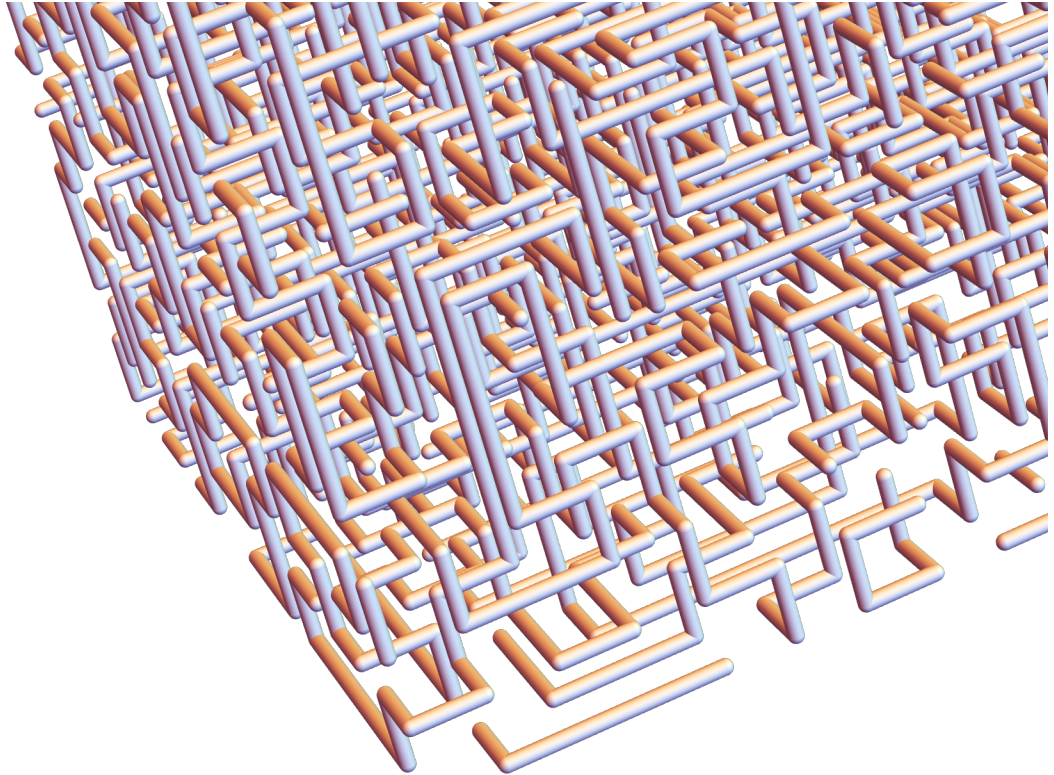
In[ ]:= n = 16;
moves = {{1, 0, 0}, {-1, 0, 0}, {0, 1, 0}, {0, -1, 0}, {0, 0, 1}, {0, 0, -1}};
visited = Table[False, {n}, {n}, {n}];
path = {}; c = 0;
While[Length[pos = Position[visited, False]] > 0,
  ++c;
  visited[[Sequence@@ (at = RandomChoice[pos])]] = True;
  dirs = Range[6];
  While[Length[dirs] > 0,
    d = RandomChoice[dirs];
    If[(And@@ ((1 ≤ # ≤ n) & /@ (next = at + moves[[d]]))) ^ ! Extract[visited, next],
      AppendTo[path, Tube[{at, at = next}]];
      visited[[Sequence@@at]] = True;
      dirs = Range[6],
      (*else*) dirs = DeleteCases[dirs, d]
    ]
  ]
];
c
Graphics3D[path, Boxed → False]

```

Out[]:= 342



Out[]:=

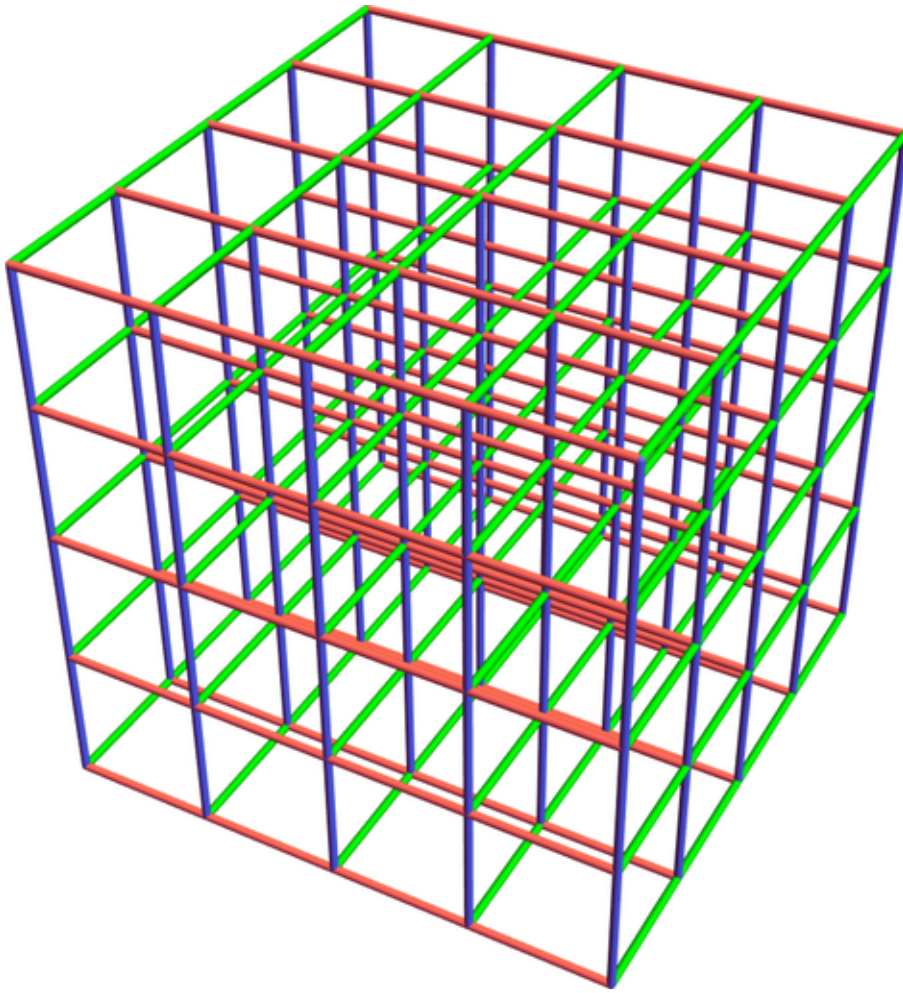


```

In[ ]:= n = 5;
{e1, e2, e3} = IdentityMatrix[3];
es = RandomSample[Join[
  (# -> # + {1, 0, 0}) & /@ Tuples[{Range[n - 1], Range[n], Range[n]}],
  (# -> # + {0, 1, 0}) & /@ Tuples[{Range[n], Range[n - 1], Range[n]}],
  (# -> # + {0, 0, 1}) & /@ Tuples[{Range[n], Range[n], Range[n - 1]}]
], Round[(1 #) & [3 n2 (n - 1)]];
MakeImage["4x4x4_Box", Graphics3D[{es /. x_ -> y_ ->
  {Abs[y - x] /. {e1 -> Pink, e2 -> Green, e3 -> RGBColor[0.5, 0.5, 1]}}, Tube[{x, y}]}],
  {Axes -> False, Boxed -> False}],
  ImageSize -> 800]

```

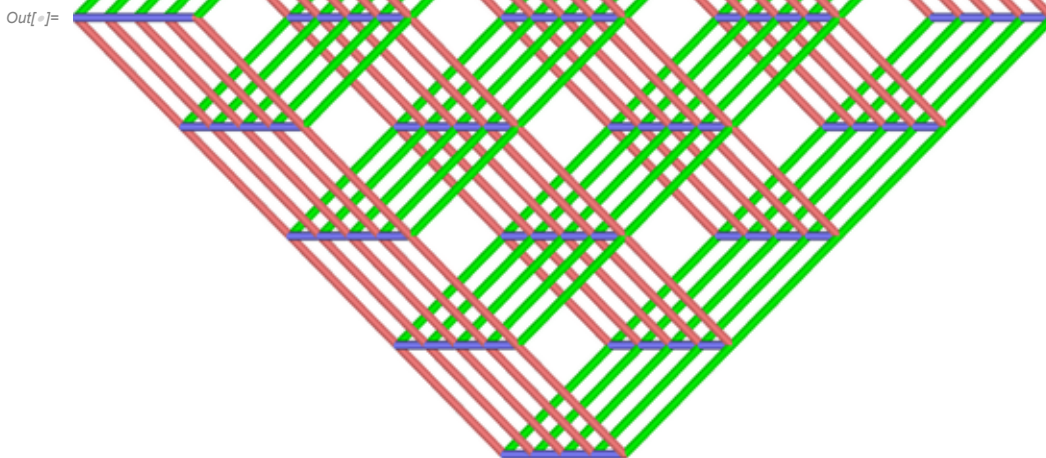
Out[]:=




```

In[ ]:= n = 5;
{e1, e2, e3} = IdentityMatrix[3];
es = RandomSample[Join[
  (# -> # + {1, 0, 0}) & /@ Tuples[{Range[n - 1], Range[n], Range[n]}],
  (# -> # + {0, 1, 0}) & /@ Tuples[{Range[n], Range[n - 1], Range[n]}],
  (# -> # + {0, 0, 1}) & /@ Tuples[{Range[n], Range[n], Range[n - 1]}]
], Round[(1 #) & [3 n^2 (n - 1)]];
MakeImage["4x4x4_Grid", Graphics3D[{es /. x_ -> y_ ->
  {Abs[y - x] /. {e1 -> Pink, e2 -> Green, e3 -> RGBColor[0.5, 0.5, 1]}, Tube[{x, y]}},
{Axes -> False, Boxed -> False, ImageSize -> {810.474, 837.}, Lighting ->
  {"Directional", White, {10, 10, 10}}, {"Directional", White, {-10, -10, 10}},
ViewPoint -> {-0.275, -0.275, 1.96107}, ViewProjection -> "Orthographic",
ViewVertical -> {-0.7, 0.7, 0}},
ImageSize ->
800]

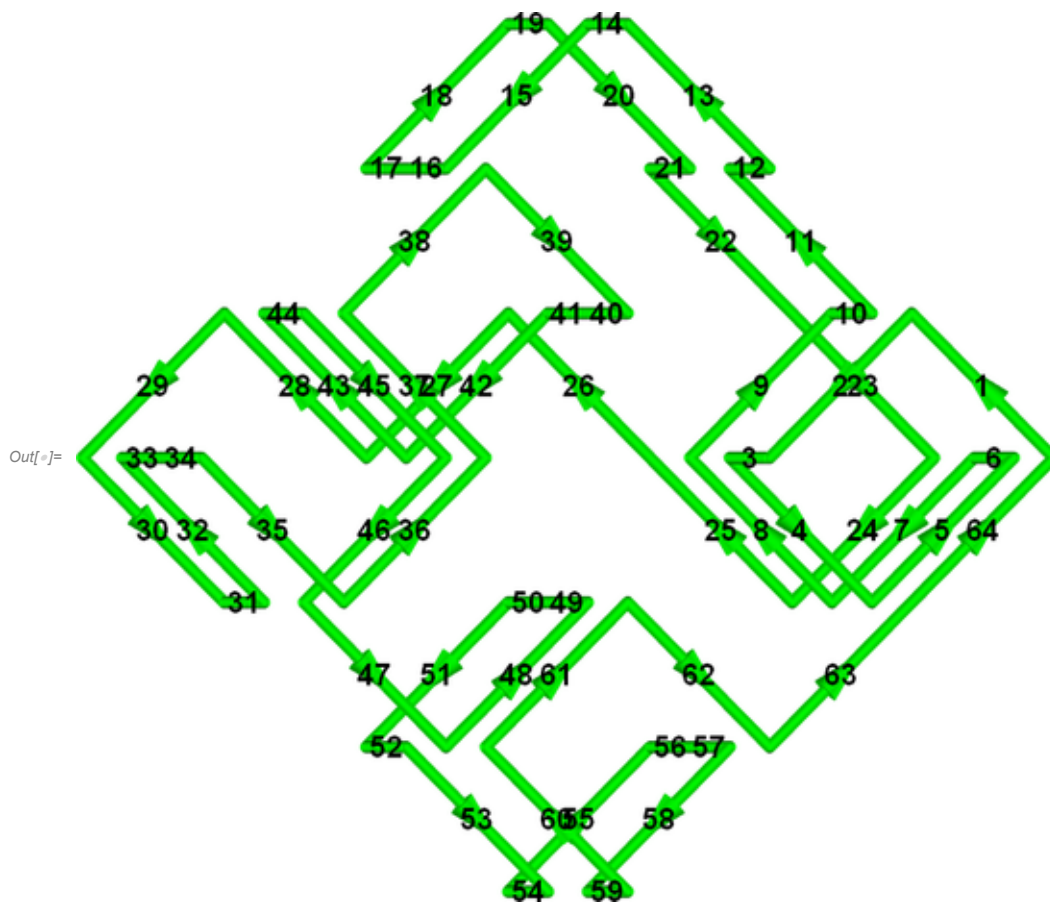
```



```

In[ ]:= n = 4; SeedRandom[3];
{e1, e2, e3} = IdentityMatrix[3];
g = Graph[
  vs = RandomSample[Tuples[Range[n], 3]],
  es = RandomSample[Join[
    (# -> # + {1, 0, 0}) & /@ Tuples[{Range[n - 1], Range[n], Range[n]}],
    (# -> # + {0, 1, 0}) & /@ Tuples[{Range[n], Range[n - 1], Range[n]}],
    (# -> # + {0, 0, 1}) & /@ Tuples[{Range[n], Range[n], Range[n - 1]}]
  ], Round[(1 #) & [3 n^2 (n - 1)]]]
];
hs = First@FindHamiltonianCycle[g];
hs = RotateLeft[hs, Position[hs, {n, n, n} -> _][[1, 1]] - 1];
(*HighlightGraph[g, PathGraph@@hs] *)
nhs = Complement[es, hs ∪ (Reverse /@ hs)];
MakeImage["Knot_in_3x3x3", Graphics3D[{Arrowheads[0.05], Green,
  hs /. x_ -> y_ -> {
    Abs[y - x] /. {e1 | e2 -> {Arrow@Tube[{x, (2 y + x) / 3}, 0.03],
      Tube[{(2 x + y) / 3, y}, 0.03]}, e3 -> Tube[{x, y}, 0.03]}
  },
  k = 0; Black,
  hs /. x_ -> y_ -> Text[Style[+, k, FontSize -> 24, FontWeight -> Bold], (x + y) / 2]
},
Boxed -> False, Axes -> False,
Lighting -> {"Directional", White, 10 {1, 1, 1}}, {"Directional", White, 10 {-1, -1, 1}},
ViewPoint -> {-0.275, -0.275, 1.96107},
ViewProjection -> "Orthographic", ViewVertical -> {-0.7, 0.7, 0}],
ImageSize -> 800]

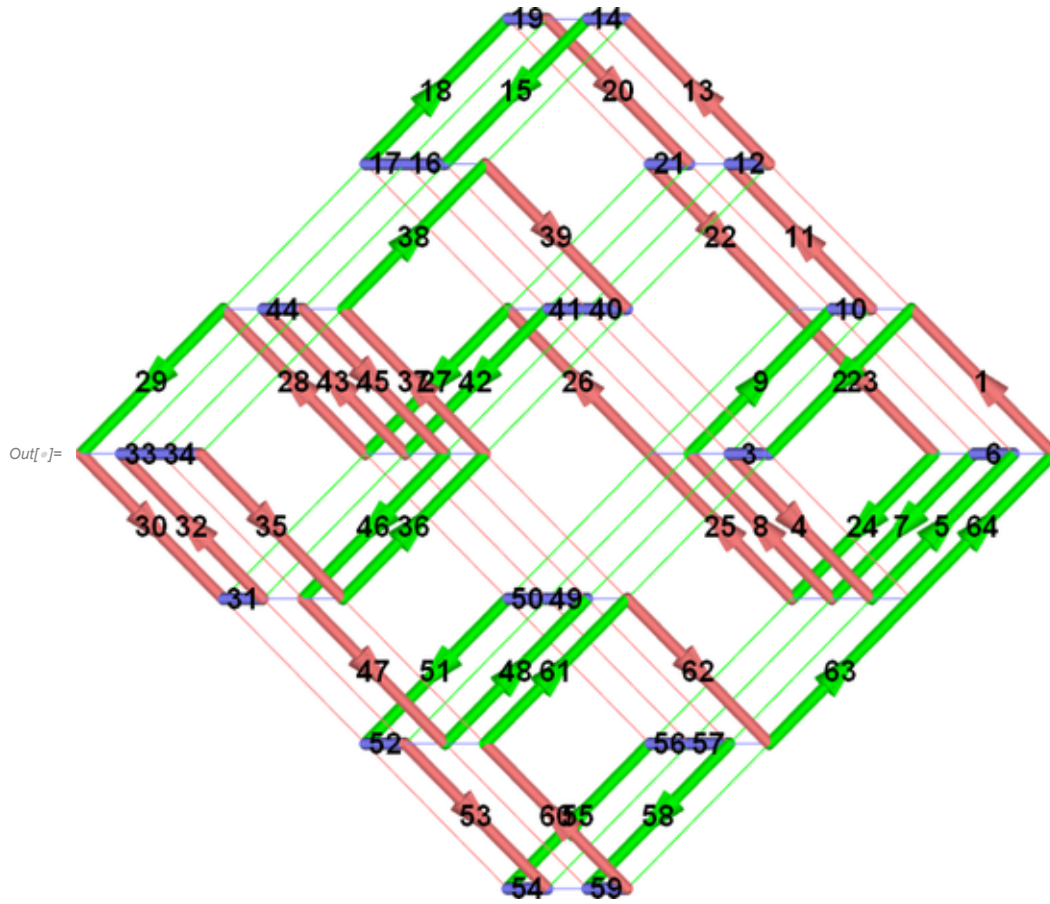
```




```

In[ ]:= n = 4; SeedRandom[3];
{e1, e2, e3} = IdentityMatrix[3];
g = Graph[
  vs = RandomSample[Tuples[Range[n], 3]],
  es = RandomSample[Join[
    (# -> # + {1, 0, 0}) & /@ Tuples[{Range[n - 1], Range[n], Range[n]}],
    (# -> # + {0, 1, 0}) & /@ Tuples[{Range[n], Range[n - 1], Range[n]}],
    (# -> # + {0, 0, 1}) & /@ Tuples[{Range[n], Range[n], Range[n - 1]}]
  ], Round[(1 #) & [3 n^2 (n - 1)]]]
];
hs = First@FindHamiltonianCycle[g];
hs = RotateLeft[hs, Position[hs, {n, n, n} -> _][[1, 1] - 1];
(*HighlightGraph[g, PathGraph@@hs]*)
nhs = Complement[es, hs | (Reverse /@ hs)];
MakeImage["Knot_in_3x3x3_with_Grid", Graphics3D[{nhs /. x_ -> y_ ->
  {Abs[y - x] /. {e1 -> Pink, e2 -> Green, e3 -> RGBColor[0.5, 0.5, 1]}, Line[{x, y}]},
  Arrowheads[0.05],
  hs /. x_ -> y_ -> {
    (t = Abs[y - x]) /. {e1 -> Pink, e2 -> Green, e3 -> RGBColor[0.5, 0.5, 1]},
    t /. {e1 | e2 -> {Arrow@Tube[{x, (2 y + x) / 3}, 0.03], Tube[{(2 x + y) / 3, y}, 0.03]},
    e3 -> Tube[{x, y}, 0.03]}
  },
  k = 0; Black,
  hs /. x_ -> y_ -> Text[Style[++k, FontSize -> 24, FontWeight -> Bold], (x + y) / 2]
],
Boxed -> False, Axes -> False,
Lighting -> {"Directional", White, 10 {1, 1, 1}}, {"Directional", White, 10 {-1, -1, 1}},
ViewPoint -> {-0.275, -0.275, 1.96107},
ViewProjection -> "Orthographic", ViewVertical -> {-0.7, 0.7, 0}],
ImageSize -> 800]

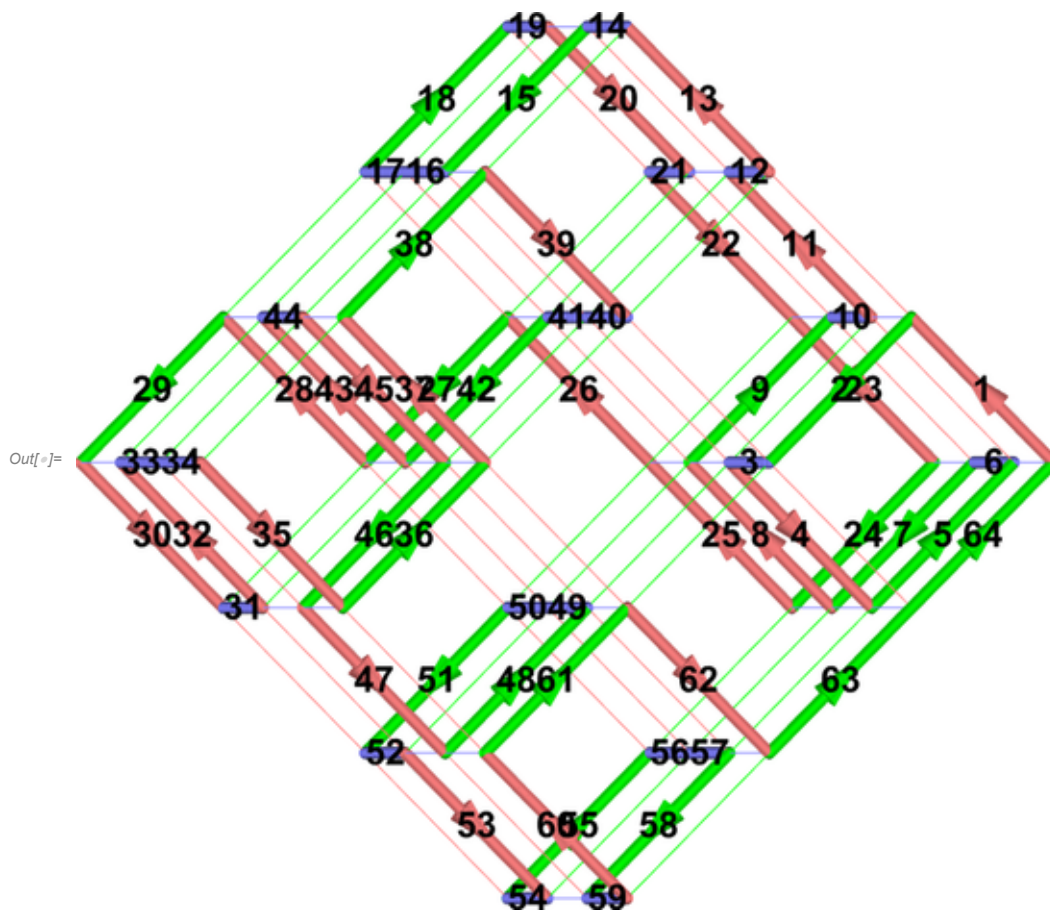
```



```

In[ ]:= n = 4; SeedRandom[3];
{e1, e2, e3} = IdentityMatrix[3];
g = Graph[
  vs = RandomSample[Tuples[Range[n], 3]],
  es = RandomSample[Join[
    (# -> # + {1, 0, 0}) & /@ Tuples[{Range[n - 1], Range[n], Range[n]}],
    (# -> # + {0, 1, 0}) & /@ Tuples[{Range[n], Range[n - 1], Range[n]}],
    (# -> # + {0, 0, 1}) & /@ Tuples[{Range[n], Range[n], Range[n - 1]}]
  ], Round[(1 #) & [3 n^2 (n - 1)]]]
];
hs = First@FindHamiltonianCycle[g];
hs = RotateLeft[hs, Position[hs, {n, n, n} -> _][[1, 1] - 1];
(*HighlightGraph[g, PathGraph@@hs]*)
nhs = Complement[es, hs | (Reverse /@ hs)];
MakeImage["Knot_in_3x3x3_with_Grid_V2", Graphics3D[{nhs /. x_ -> y_ ->
  {Abs[y - x] /. {e1 -> Pink, e2 -> Green, e3 -> RGBColor[0.5, 0.5, 1]}, Line[{x, y}]},
  Arrowheads[0.05],
  hs /. x_ -> y_ -> {
    (t = Abs[y - x]) /. {e1 -> Pink, e2 -> Green, e3 -> RGBColor[0.5, 0.5, 1]},
    t /. {e1 | e2 -> {Arrow@Tube[{x, (2 y + x) / 3}, 0.03], Tube[{(2 x + y) / 3, y}, 0.03]},
    e3 -> Tube[{x, y}, 0.03]}
  },
  k = 0; Black,
  hs /. x_ -> y_ -> Text[Style[++k, FontSize -> 28, FontWeight -> Bold], (x + y) / 2]
],
Boxed -> False, Axes -> False,
Lighting -> {"Directional", White, 10 {1, 1, 1}}, {"Directional", White, 10 {-1, -1, 1}},
ViewPoint -> {-0.275, -0.275, 1.96107},
ViewProjection -> "Orthographic", ViewVertical -> {-0.7, 0.7, 0},
ImageSize -> 800]

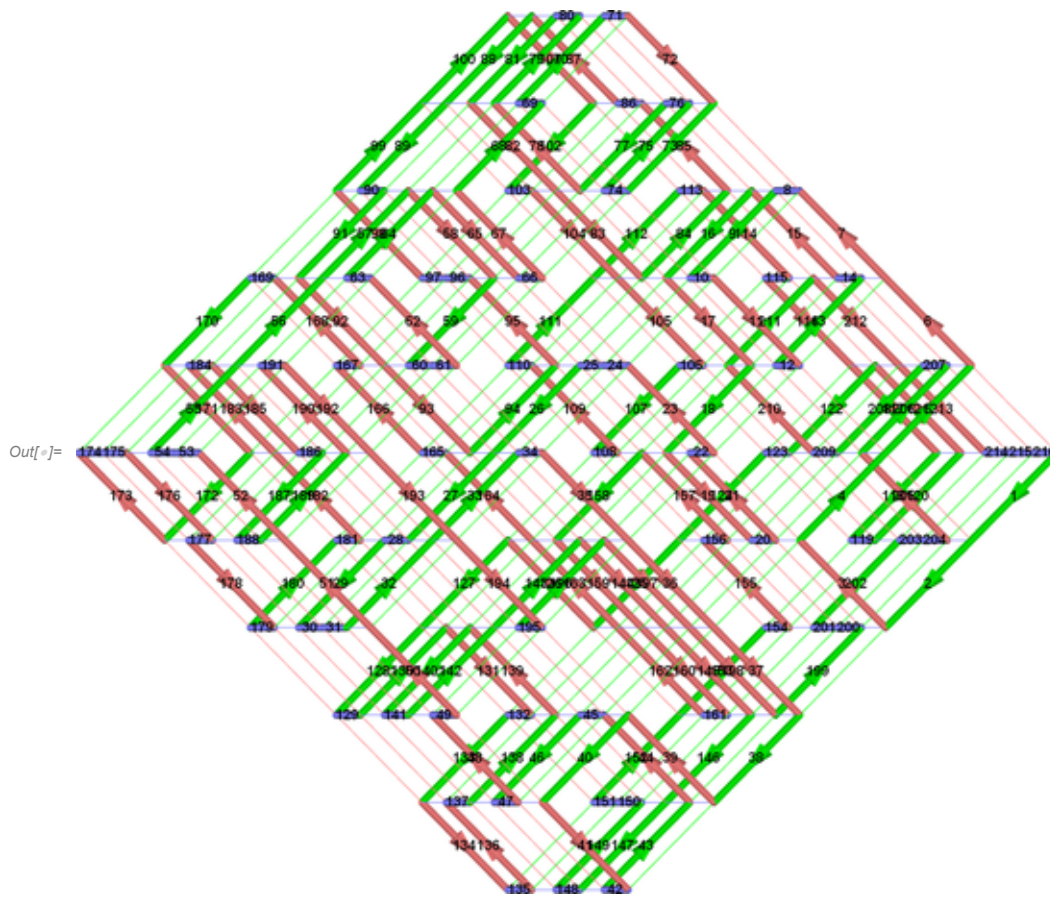
```



```

In[ ]:= n = 6; SeedRandom[3];
{e1, e2, e3} = IdentityMatrix[3];
g = Graph[
  vs = RandomSample[Tuples[Range[n], 3]],
  es = RandomSample[Join[
    (# -> # + {1, 0, 0}) & /@ Tuples[{Range[n - 1], Range[n], Range[n]}],
    (# -> # + {0, 1, 0}) & /@ Tuples[{Range[n], Range[n - 1], Range[n]}],
    (# -> # + {0, 0, 1}) & /@ Tuples[{Range[n], Range[n], Range[n - 1]}]
  ], Round[(1 #) & [3 n^2 (n - 1)]]]
];
hs = First@FindHamiltonianCycle[g];
hs = RotateLeft[hs, Position[hs, {n, n, n} -> _][[1, 1] - 1];
(*HighlightGraph[g, PathGraph@hs]*)
nhs = Complement[es, hs | (Reverse /@ hs)];
MakeImage["Knot_in_5x5x5_with_Grid", Graphics3D[{nhs /. x_ -> y_ ->
  {Abs[y - x] /. {e1 -> Pink, e2 -> Green, e3 -> RGBColor[0.5, 0.5, 1]}, Line[{x, y}]},
  Arrowheads[0.03],
  hs /. x_ -> y_ -> {
    (t = Abs[y - x]) /. {e1 -> Pink, e2 -> Green, e3 -> RGBColor[0.5, 0.5, 1]},
    t /. {e1 | e2 -> {Arrow@Tube[{x, (2 y + x) / 3}, 0.03], Tube[{(2 x + y) / 3, y}, 0.03]},
    e3 -> Tube[{x, y}, 0.03]}
  },
  k = 0; Black,
  hs /. x_ -> y_ -> Text[Style[++k, FontSize -> 16, FontWeight -> Bold], (x + y) / 2]
],
{Axes -> False, Boxed -> False, ImageSize -> {810.474, 837.}, Lighting ->
  {"Directional", White, {10, 10, 10}}, {"Directional", White, {-10, -10, 10}}},
ViewPoint -> {-0.275, -0.275, 1.96107}, ViewProjection -> "Orthographic",
ViewVertical -> {-0.7, 0.7, 0}}],
ImageSize ->
1200]

```




```

In[ ]:= n = 6; SeedRandom[3];
{e1, e2, e3} = IdentityMatrix[3];
g = Graph[
  vs = RandomSample[Tuples[Range[n], 3]],
  es = RandomSample[Join[
    (# -> # + {1, 0, 0}) & /@ Tuples[{Range[n - 1], Range[n], Range[n]}],
    (# -> # + {0, 1, 0}) & /@ Tuples[{Range[n], Range[n - 1], Range[n]}],
    (# -> # + {0, 0, 1}) & /@ Tuples[{Range[n], Range[n], Range[n - 1]}]
  ], Round[(1 #) & [3 n^2 (n - 1)]]]
];
hs = First@FindHamiltonianCycle[g];
hs = RotateLeft[hs, Position[hs, {n, n, n} -> _][[1, 1]] - 1];
(*HighlightGraph[g, PathGraph@hs] *)
nhs = Complement[es, hs | (Reverse /@ hs)];
MakeImage["Knot_in_5x5x5_with_Grid_V2",
Graphics3D[{nhs /. x_ -> y_ -> {Abs[y - x] /. {e1 -> RGBColor[1, 0.75, 0.75],
  e2 -> RGBColor[0.75, 1, 0.75], e3 -> RGBColor[0.75, 0.75, 1]}, Line[{x, y]}},
Arrowheads[0.03],
hs /. x_ -> y_ -> {
  (t = Abs[y - x]) /. {e1 -> Pink, e2 -> Green, e3 -> RGBColor[0.5, 0.5, 1]},
  t /. {e1 | e2 -> {Arrow@Tube[{x, (2 y + x) / 3}, 0.03], Tube[{(2 x + y) / 3, y}, 0.03]},
  e3 -> Tube[{x, y}, 0.03]}
},
k = 0; Black,
hs /. x_ -> y_ -> Text[Style[++k, FontSize -> 20, FontWeight -> Bold], (x + y) / 2]
},
{Axes -> False, Boxed -> False, ImageSize -> {810.474, 837.}, Lighting ->
  {"Directional", White, {10, 10, 10}}, {"Directional", White, {-10, -10, 10}}},
ViewPoint -> {-0.275, -0.275, 1.96107}, ViewProjection -> "Orthographic",
ViewVertical -> {-0.7, 0.7, 0}}],
ImageSize ->
1200]

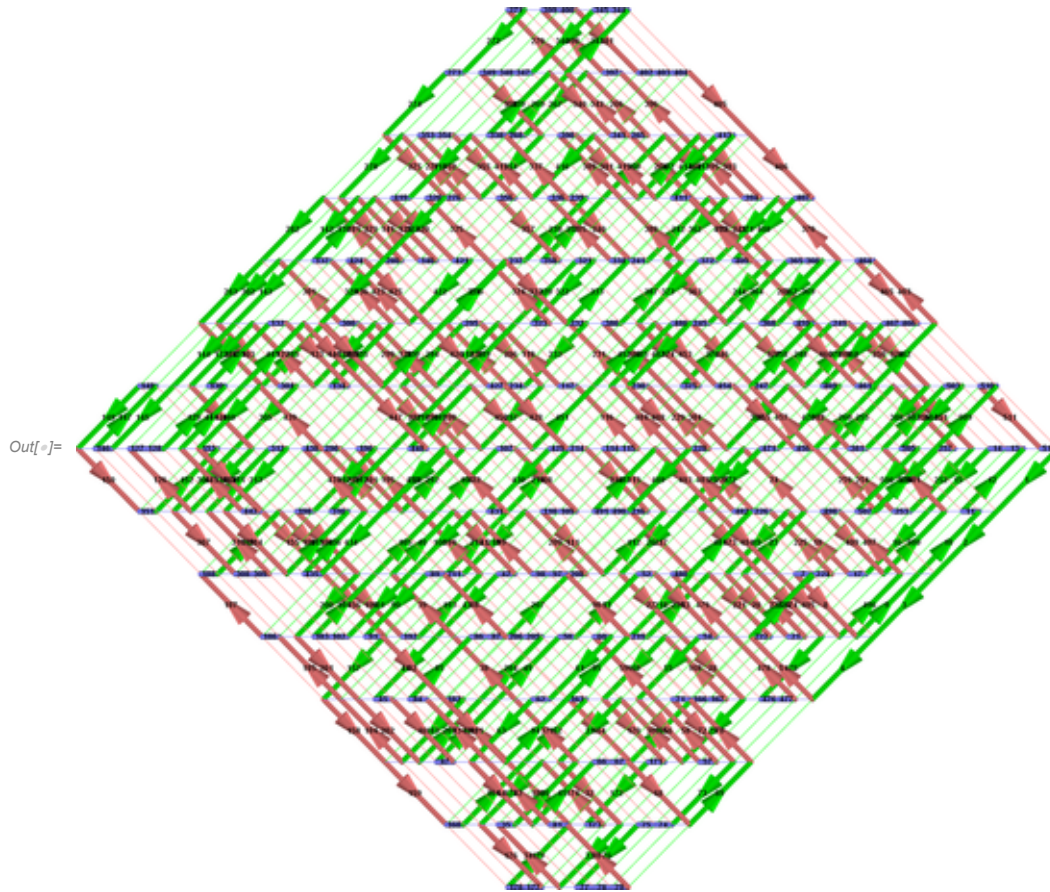
```



```

(*n=8; SeedRandom[6];
{e1,e2,e3}=IdentityMatrix[3];
g=Graph[
  vs=RandomSample[Tuples[Range[n],3]],
  es=RandomSample[Join[
    {#->#+{1,0,0}&/@Tuples[{Range[n-1],Range[n],Range[n]}],
    {#->#+{0,1,0}&/@Tuples[{Range[n],Range[n-1],Range[n]}],
    {#->#+{0,0,1}&/@Tuples[{Range[n],Range[n],Range[n-1]}]
  ],Round[(1#)&[3n^2(n-1)]]
];
hs=First@FindHamiltonianCycle[g];
hs=RotateLeft[hs,Position[hs,{n,n,n}<->_][[1,1]]-1];
(*HighlightGraph[g,PathGraph@@hs]*)
nhs=Complement[es,hs∪(Reverse/@hs)];
MakeImage["Knot_in_7x7x7_with_Grid",Graphics3D[
  {nhs/.x_<->y_>:{Abs[y-x]/.{e1->Pink,e2->Green,e3->RGBColor[0.5,0.5,1]},Line[{x,y]}},
  Arrowheads[0.03],
  hs/.x_<->y_>:{
    (t=Abs[y-x])/.{e1->Pink,e2->Green,e3->RGBColor[0.5,0.5,1]},
    t/.{e1|e2->{Arrow@Tube[{x,(2y+x)/3},0.03],Tube[{(2x+y)/3,y},0.03]},
    e3->Tube[{x,y},0.03]}
  },
  k=0; Black,
  hs/.x_<->y_>Text[Style[#+k,FontSize->12,FontWeight->Bold],(x+y)/2]
],
{Axes->False,Boxed->False,ImageSize->{810.474,837.},
  Lighting->{{"Directional",White,{10,10,10}},{ "Directional",White,{-10,-10,10}}},
  ViewPoint->{-0.275,-0.275,1.96107},
  ViewProjection->"Orthographic",ViewVertical->{-0.7,0.7,0}},
ImageSize->1600]*)

```



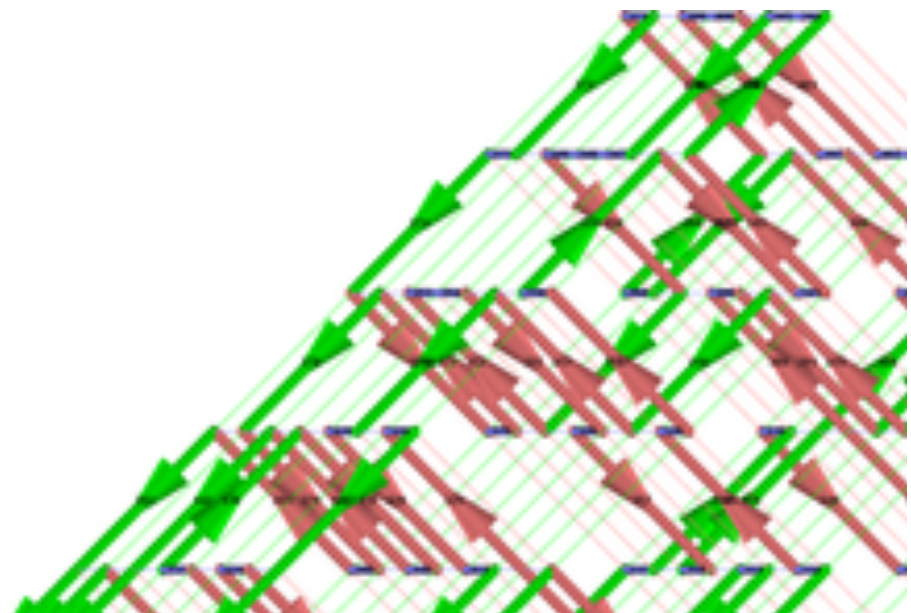
Out[]=

(*hs>>HamiltonCycleIn7x7x7.m*)

```

In[ ]:= n = 8;
{e1, e2, e3} = IdentityMatrix[3];
g = Graph[
  vs = Tuples[Range[n], 3],
  es = Join[
    (# -> # + {1, 0, 0}) & /@ Tuples[{Range[n - 1], Range[n], Range[n]}],
    (# -> # + {0, 1, 0}) & /@ Tuples[{Range[n], Range[n - 1], Range[n]}],
    (# -> # + {0, 0, 1}) & /@ Tuples[{Range[n], Range[n], Range[n - 1]}]
  ]
];
hs = Get["HamiltonCycleIn7x7x7.m"];
(*HighlightGraph[g, PathGraph@@hs]*)
nhs = Complement[es, hs ∪ (Reverse /@ hs)];
MakeImage["Knot_in_7x7x7_with_Grid", Graphics3D[{nhs /. x_ -> y_ ->
  {Abs[y - x] /. {e1 -> Pink, e2 -> Green, e3 -> RGBColor[0.5, 0.5, 1]}, Line[{x, y]}},
  Arrowheads[0.03],
  hs /. x_ -> y_ -> {
    (t = Abs[y - x]) /. {e1 -> Pink, e2 -> Green, e3 -> RGBColor[0.5, 0.5, 1]},
    t /. {e1 | e2 -> {Arrow@Tube[{x, (2 y + x) / 3}, 0.03], Tube[{(2 x + y) / 3, y}, 0.03]}},
    e3 -> Tube[{x, y}, 0.02]}
  },
  k = 0; Black,
  hs /. x_ -> y_ -> Text[Style[++k, FontSize -> 12, FontWeight -> Bold], (x + y) / 2]
],
{Axes -> False, Boxed -> False, ImageSize -> {810.474, 837.}, Lighting ->
  {"Directional", White, {10, 10, 10}}, {"Directional", White, {-10, -10, 10}},
  ViewPoint -> {-0.205, -0.205, 1.96107}, ViewProjection -> "Orthographic",
  ViewVertical -> {-0.7, 0.7, 0}},
ImageSize ->
  2400]

```



Out[]=

