State Capital Tours

Mark Senn

February 28, 2011

The state capital tours problem is to find the shortest and longest tour lengths for visiting all state capitals in the lower 48 states. Each solution starts at any state capital and then visits each of the remaining 47 state capitals once.

I asked four people to guess at solutions. Short trip guesses were 6,000 to 75,000 miles; long trip guesses were 100,000 to "more than 250,000 miles."

The *Mathematica* program below solves this problem. It doesn't try every permutation of states but tries to give a good answer. The program took 71 seconds (update this time this after all modifications to program done) to run on a 8 CPU Intel Core i7 920 running at 2.67 GHz using *Mathematica* 8.0.0.0 on Fedora 14 Linux. The default *Mathematica* method and ten user-chosen methods were used to find each tour. The shortest tour found was 10,648.5 miles; the longest tour found was 70,587.1 miles. The default *Mathematica* method did not find the shortest or longest tour.

In[143]:=
```
(* Define PrintTourInformation. *)
PrintTourInformation[heading_, caplatlong_, tour_] := Module[
   {print, t},

  (* Print heading. *)
  Print[Graphics[{LightGray, Rectangle[{0, 0}, {1, 0.25}]}],
   Style[" " <> heading, 12]];

  (* Print the tour distances and order of states for tour. *)
  print = tour;
  (* Wolfram|Alpha stated there are 125/201168 meters/mile. *)
  print[[All, 1]] = Abs[# * 125 / 201168] & /@ print[[All, 1]];
  print[[All, 2, All]] = table[[#, 2]] & /@ print[[All, 2, All]];
  Print[print];

  (* Change caplatlong from
   (latitude,longitude) order to (x,y) order for plotting. *)
  t = Reverse[#] & /@ caplatlong;

  (* Print the tour distances and plot of tour. *)
  Print[
   Table[
    {
     print[[i, 1]],
     ListLinePlot[
      t[[tour[[i, 2]]]],
      Axes → False
     ]
    },
    {i, 1, Length[tour]}
   ]
  ]
 ]
```

In[144]:=
```
(* Save starting time. *)
starttime = AbsoluteTime[];
```

I got the following *Mathematica* code by typing "==United States state abbreviations" (that's two equal signs) in an input cell. In the resulting "Table:" box, the third box down, I left clicked on the + with a gray circle around it and chose "Computable data". I then deleted the input and output cells from the "==United States state abbreviations" input.

```
In[145]:= (* Get state abbreviations. *)
          ab = WolframAlpha[
              "United States state abbreviations",
              {{"Table:Abbreviation:USStateData", 1}, "ComputableData"}
            ];
          ab
```

```
Out[146]= {{Arizona, AZ}, {California, CA}, {Georgia, GA}, {Indiana, IN}, {Montana, MT},
           {Ohio, OH}, {Virginia, VA}, {Kansas, KS}, {Massachusetts, MA}, {Nebraska, NE},
           {Oklahoma, OK}, {Alaska, AK}, {SouthDakota, SD}, {Hawaii, HI}, {Alabama, AL},
           {Arkansas, AR}, {Colorado, CO}, {Connecticut, CT}, {Delaware, DE},
           {Florida, FL}, {Idaho, ID}, {Illinois, IL}, {Iowa, IA}, {Kentucky, KY},
           {Louisiana, LA}, {Maine, ME}, {Maryland, MD}, {Michigan, MI}, {Minnesota, MN},
           {Mississippi, MS}, {Missouri, MO}, {Nevada, NV}, {NewHampshire, NH},
           {NewJersey, NJ}, {NewMexico, NM}, {NewYork, NY}, {NorthCarolina, NC},
           {NorthDakota, ND}, {Oregon, OR}, {Pennsylvania, PA}, {RhodeIsland, RI},
           {SouthCarolina, SC}, {Tennessee, TN}, {Texas, TX}, {Utah, UT}, {Vermont, VT},
           {Washington, WA}, {WestVirginia, WV}, {Wisconsin, WI}, {Wyoming, WY}}
```

I got the following *Mathematica* code by typing "==United States state capitals" (that's two equal signs) in an input cell. In the resulting "Table:" box, the fourth box down, I left clicked on "More" until "More" went away, then left clicked the + with a gray circle around it and chose "Computable data". I then deleted the input and output cells from the "==United States state capitals" input.

```
In[147]:= cap = WolframAlpha[
            "United States state capitals",
            {{"Table:CapitalCity:USStateData", 1}, "ComputableData"},
            PodStates → {
              "Table:CapitalCity:USStateData__More",
              "Table:CapitalCity:USStateData__More",
              "Table:CapitalCity:USStateData__More"
             }
           ]
```

```
Out[147]= {{Arizona, {Phoenix, Arizona, UnitedStates}},
           {California, {Sacramento, California, UnitedStates}},
           {Georgia, {Atlanta, Georgia, UnitedStates}},
           {Indiana, {Indianapolis, Indiana, UnitedStates}},
           {Montana, {Helena, Montana, UnitedStates}}, {Ohio, {Columbus, Ohio, UnitedStates}},
           {Virginia, {Richmond, Virginia, UnitedStates}},
           {Kansas, {Topeka, Kansas, UnitedStates}},
           {Massachusetts, {Boston, Massachusetts, UnitedStates}},
           {Nebraska, {Lincoln, Nebraska, UnitedStates}},
           {Oklahoma, {OklahomaCity, Oklahoma, UnitedStates}},
           {Alaska, {Juneau, Alaska, UnitedStates}},
           {SouthDakota, {Pierre, SouthDakota, UnitedStates}},
           {Hawaii, {Honolulu, Hawaii, UnitedStates}},
           {Alabama, {Montgomery, Alabama, UnitedStates}},
           {Arkansas, {LittleRock, Arkansas, UnitedStates}},
```

```
{Colorado, {Denver, Colorado, UnitedStates}},
{Connecticut, {Hartford, Connecticut, UnitedStates}},
{Delaware, {Dover, Delaware, UnitedStates}},
{Florida, {Tallahassee, Florida, UnitedStates}},
{Idaho, {BoiseCity, Idaho, UnitedStates}},
{Illinois, {Springfield, Illinois, UnitedStates}},
{Iowa, {DesMoines, Iowa, UnitedStates}},
{Kentucky, {Frankfort, Kentucky, UnitedStates}},
{Louisiana, {BatonRouge, Louisiana, UnitedStates}},
{Maine, {Augusta, Maine, UnitedStates}},
{Maryland, {Annapolis, Maryland, UnitedStates}},
{Michigan, {Lansing, Michigan, UnitedStates}},
{Minnesota, {SaintPaul, Minnesota, UnitedStates}},
{Mississippi, {Jackson, Mississippi, UnitedStates}},
{Missouri, {JeffersonCity, Missouri, UnitedStates}},
{Nevada, {Carson, Nevada, UnitedStates}},
{NewHampshire, {Concord, NewHampshire, UnitedStates}},
{NewJersey, {Trenton, NewJersey, UnitedStates}},
{NewMexico, {SantaFe, NewMexico, UnitedStates}},
{NewYork, {Albany, NewYork, UnitedStates}},
{NorthCarolina, {Raleigh, NorthCarolina, UnitedStates}},
{NorthDakota, {Bismarck, NorthDakota, UnitedStates}},
{Oregon, {Salem, Oregon, UnitedStates}},
{Pennsylvania, {Harrisburg, Pennsylvania, UnitedStates}},
{RhodeIsland, {Providence, RhodeIsland, UnitedStates}},
{SouthCarolina, {Columbia, SouthCarolina, UnitedStates}},
{Tennessee, {Nashville, Tennessee, UnitedStates}},
{Texas, {Austin, Texas, UnitedStates}}, {Utah, {SaltLakeCity, Utah, UnitedStates}},
{Vermont, {Montpelier, Vermont, UnitedStates}},
{Washington, {Olympia, Washington, UnitedStates}},
{WestVirginia, {Charleston, WestVirginia, UnitedStates}},
{Wisconsin, {Madison, Wisconsin, UnitedStates}},
{Wyoming, {Cheyenne, Wyoming, UnitedStates}}}
```

In[148]:=
```
(* Get the state capitals' coordinates. *)

(*
  This statement:
    latlong =
 Function[x, CityData[x,"Coordinates"]] /@ Map[ToString, cap[[All,2]], {2}];
  can be shortened to
    latlong = CityData[#,"Coordinates"]&/@ Map[ToString, cap[[All,2]], {2}];
  which _can't_ be shortened to
    latlong = CityData[#,"Coordinates"]&/@ ToString, {2}&/@ cap[[All,2]]
*)
latlong = CityData[#, "Coordinates"] & /@ Map[ToString, cap[[All, 2]], {2}];
```

In[149]:=
```
(* Put number, abbreviation, capital,
latitude, and longitude information in data. *)
data = {ab[[All, 2]], ab[[All, 1]], cap[[All, 2, 1]], latlong};
data = Transpose[data];
```

In[151]:=
```
(*
 Get data for only the continental United States and
 sort by state abbreviation the first column of each row.
*)
lower48 = Sort[
   Select[
    data,
    # [[2]] ≠ "Alaska" && #[[2]] ≠ "Hawaii" &
   ]
  ];
```

In[152]:=
```
(*
  Change, for example,
"NorthCaroline" and "SaltLakeCity" to "North Carolina" and "Salt Lake City".
*)
Table[
  lower48[[All, i]]
   = StringReplace[
      #,
      RegularExpression["([[:lower:]])([[:upper:]])"] → "$1 $2"
     ]
     & /@ lower48[[All, i]],
  {i, 2, 3}
 ];
```

In[153]:=
```
(* Construct the table of information to print. *)
table = Flatten[#] & /@ lower48;
Table[
  table[[i]] = Prepend[table[[i]], i],
  {i, 1, Length[lower48]}
 ];
```

In[155]:=
```
(* Print the table of information. *)
Grid[
 table,
 Alignment → {{Right, Left, Left, Left, ".", "."}},
 Spacings → {2, 0.3}
]
```

Out[155]=

1	AL	Alabama	Montgomery	32.3615	−86.2791
2	AR	Arkansas	Little Rock	34.736	−92.3311
3	AZ	Arizona	Phoenix	33.5284	−112.076
4	CA	California	Sacramento	38.5556	−121.469
5	CO	Colorado	Denver	39.7263	−104.965
6	CT	Connecticut	Hartford	41.7626	−72.6886
7	DE	Delaware	Dover	39.1619	−75.5268
8	FL	Florida	Tallahassee	30.4518	−84.2728
9	GA	Georgia	Atlanta	33.7595	−84.4032
10	IA	Iowa	Des Moines	41.5909	−93.6209
11	ID	Idaho	Boise City	43.6137	−116.238
12	IL	Illinois	Springfield	39.7833	−89.6504
13	IN	Indiana	Indianapolis	39.7909	−86.1477
14	KS	Kansas	Topeka	39.0392	−95.6895
15	KY	Kentucky	Frankfort	38.1973	−84.8631
16	LA	Louisiana	Baton Rouge	30.4581	−91.1402
17	MA	Massachusetts	Boston	42.3216	−71.0891
18	MD	Maryland	Annapolis	38.9729	−76.5012
19	ME	Maine	Augusta	44.3235	−69.7653
20	MI	Michigan	Lansing	42.7176	−84.5549
21	MN	Minnesota	Saint Paul	44.9544	−93.1141
22	MO	Missouri	Jefferson City	38.573	−92.1893
23	MS	Mississippi	Jackson	32.3204	−90.2044
24	MT	Montana	Helena	46.5958	−112.027
25	NC	North Carolina	Raleigh	35.8188	−78.6446
26	ND	North Dakota	Bismarck	46.8133	−100.779
27	NE	Nebraska	Lincoln	40.8099	−96.6753
28	NH	New Hampshire	Concord	43.2201	−71.5491
29	NJ	New Jersey	Trenton	40.2217	−74.7561
30	NM	New Mexico	Santa Fe	35.6672	−105.965
31	NV	Nevada	Carson	39.1609	−119.754
32	NY	New York	Albany	42.6598	−73.7813
33	OH	Ohio	Columbus	39.9898	−82.9915
34	OK	Oklahoma	Oklahoma City	35.4823	−97.535
35	OR	Oregon	Salem	44.9311	−123.029
36	PA	Pennsylvania	Harrisburg	40.2698	−76.8756
37	RI	Rhode Island	Providence	41.8236	−71.4221
38	SC	South Carolina	Columbia	34.0171	−81.0108
39	SD	South Dakota	Pierre	44.368	−100.336
40	TN	Tennessee	Nashville	36.1548	−86.7621
41	TX	Texas	Austin	30.3005	−97.7472
42	UT	Utah	Salt Lake City	40.7547	−111.893
43	VA	Virginia	Richmond	37.5383	−77.4615
44	VT	Vermont	Montpelier	44.2614	−72.5728
45	WA	Washington	Olympia	47.0424	−122.893
46	WI	Wisconsin	Madison	43.0746	−89.3948
47	WV	West Virginia	Charleston	38.3495	−81.6333
48	WY	Wyoming	Cheyenne	41.1455	−104.802

In[156]:=
```
(* Get state capitals' coordinates from the sorted list. *)
caplatlong = table[[All, {5, 6}]];
```

In[157]:=
```
(* For this data, if no Method option is speciied,
the "OrOpt" method is used. *)
method = {
   "CCA",
   "Greedy",
   "GreedyCycle",
   "IntegerLinearProgramming",
   "OrOpt",
   "OrZweig",
   "RemoveCrossings",
   "SpaceFillingCurve",
   "SimulatedAnnealing",
   "TwoOpt"
  };
```

In[158]:=
```
(* Using a font size of 24 would be better. *)Print[
  Graphics[{LightGray, Rectangle[{0, 0}, {1, 0.5}]}], Style[" BEGIN OUTPUT", 12]];

(*
    Get shortest and longest tours:
        sign = 1 for shortest tour
        sign = -1 for longest tour

    The shortest tour is found my minimizing
   the sum of distances from one state capital to another.

    The longest tour is found my minimizing the sum
   of the negative distances from one state capital to another.
*)
For[sign = 1, sign ≥ -1, sign -= 2,

(* Print "SHORTEST TOUR" or "LONGEST TOUR" heading. *)
 t = "SHORTEST TOURS";
 If[sign == -1,
  t = "LONGEST TOURS"
 ];
 (* Using a font size of 18 would be better. *)
 Print[Graphics[{LightGray, Rectangle[{0, 0}, {1, 0.375}]}], Style[" " <> t, 12]];

(* Define distance function for the distance between two points. *)
 distfun[{lat1_, long1_}, {lat2_, long2_}] :=
  sign GeoDistance[{lat1, long1}, {lat2, long2}];

 (* Find the shortest or longest tour with Mathemtica choosing the method. *)
 tour = {FindShortestTour[caplatlong, DistanceFunction → distfun]};
 PrintTourInformation[
  "Mathematica found with no method specified:", caplatlong, tour];

 (* Find the shortest or longest tour using all the methods defined earlier. *)
 tour = Table[
   FindShortestTour[caplatlong, DistanceFunction → distfun, Method → method[[i]]],
   {i, 1, Length[method]}
  ];
 PrintTourInformation[" Trying ten different methods:", caplatlong, tour];

]

(* Using a font size of 24 would be better. *)
Print[Graphics[{LightGray, Rectangle[{0, 0}, {1, 0.5}]}], Style[" END OUTPUT", 12]];
```

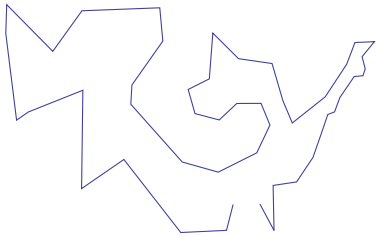
BEGIN OUTPUT

SHORTEST TOURS

Mathematica found with no method specified:

```
{{10 894.9, {AL, FL, GA, SC, NC, VA, MD, DE, NJ, CT, RI, MA,
    NH, ME, VT, NY, PA, WV, OH, MI, WI, MN, IA, NE, KS, MO, IL, IN, KY, TN,
    AR, OK, CO, WY, SD, ND, MT, ID, WA, OR, CA, NV, UT, AZ, NM, TX, LA, MS}}}
```

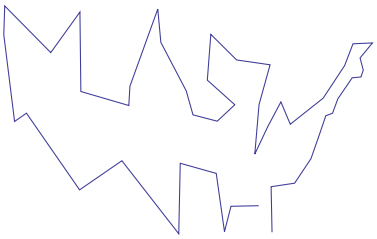
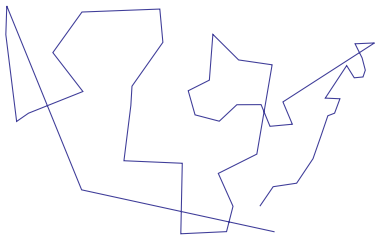
{{10 894.9,

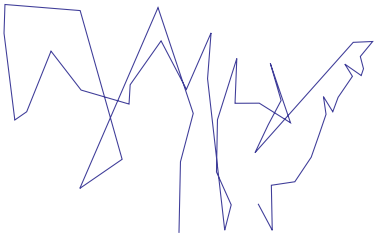
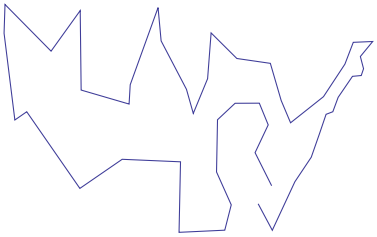


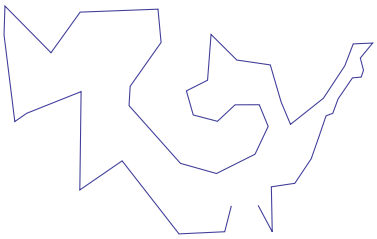
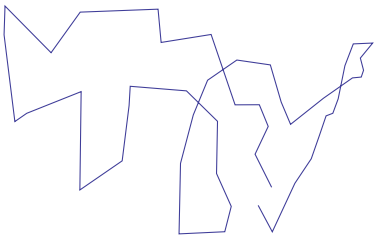
}}

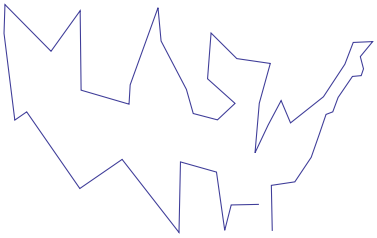
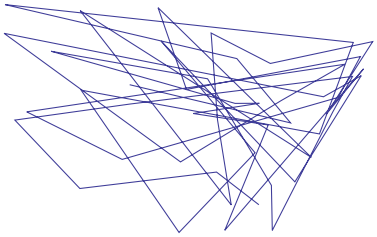
Trying ten different methods:

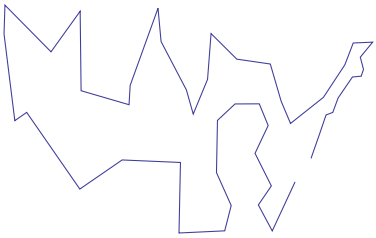
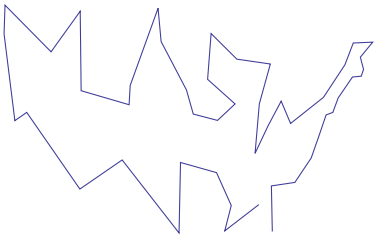
```
{{10 834.1, {AL, MS, LA, AR, OK, TX, NM, AZ, NV, CA, OR, WA,
    ID, MT, UT, CO, WY, ND, SD, NE, KS, MO, IL, IA, MN, WI, MI, IN, TN, KY,
    OH, WV, PA, NY, VT, ME, NH, MA, RI, CT, NJ, DE, MD, VA, NC, SC, GA, FL}},
 {12 860.9, {AL, GA, SC, NC, VA, MD, DE, NJ, PA, NY, CT, RI, MA, NH, VT,
    ME, OH, WV, KY, IN, IL, MO, KS, NE, IA, MN, WI, MI, TN, AR, MS, LA,
    TX, OK, NM, CO, WY, SD, ND, MT, ID, UT, NV, CA, OR, WA, AZ, FL}},
 {13 998.2, {AL, FL, GA, SC, NC, VA, MD, PA, DE, NJ, CT, NY, RI, MA, NH, ME,
    VT, TN, KY, OH, MI, WV, IN, IL, WI, MO, AR, MS, LA, IA, MN, NE,
    SD, WY, CO, UT, ID, NV, CA, OR, WA, MT, NM, AZ, ND, KS, OK, TX}},
 {10 648.5, {AL, FL, SC, NC, VA, MD, DE, NJ, CT, RI, MA, NH, ME, VT, NY, PA,
    WV, OH, MI, WI, MN, IA, KS, NE, SD, ND, WY, CO, UT, MT, ID, WA,
    OR, CA, NV, AZ, NM, OK, TX, LA, MS, AR, MO, IL, IN, KY, TN, GA}},
 {10 894.9, {AL, FL, GA, SC, NC, VA, MD, DE, NJ, CT, RI, MA, NH, ME, VT, NY,
    PA, WV, OH, MI, WI, MN, IA, NE, KS, MO, IL, IN, KY, TN, AR, OK,
    CO, WY, SD, ND, MT, ID, WA, OR, CA, NV, UT, AZ, NM, TX, LA, MS}},
 {11 075.7, {AL, FL, SC, NC, VA, MD, DE, NJ, NY, VT, ME, NH, MA, RI, CT, PA,
    WV, OH, MI, WI, IA, KS, OK, TX, LA, MS, AR, MO, NE, WY, CO, NM,
    AZ, UT, NV, CA, OR, WA, ID, MT, ND, SD, MN, IL, IN, KY, TN, GA}},
 {10 834.1, {AL, MS, LA, AR, OK, TX, NM, AZ, NV, CA, OR, WA, ID, MT, UT, CO,
    WY, ND, SD, NE, KS, MO, IL, IA, MN, WI, MI, IN, TN, KY, OH, WV,
    PA, NY, VT, ME, NH, MA, RI, CT, NJ, DE, MD, VA, NC, SC, GA, FL}},
 {46 865.4, {WY, WV, WI, WA, VT, VA, UT, TX, TN, SD, SC, RI, PA, OR, OK, OH,
    NY, NV, NM, NJ, NH, NE, ND, NC, MT, MS, MO, MN, MI, ME, MD, MA,
    LA, KY, KS, IN, IL, ID, IA, GA, FL, DE, CT, CO, CA, AZ, AR, AL}},
 {10 648.5, {NC, VA, MD, DE, NJ, CT, RI, MA, NH, ME, VT, NY, PA, WV, OH, MI,
    WI, MN, IA, KS, NE, SD, ND, WY, CO, UT, MT, ID, WA, OR, CA, NV,
    AZ, NM, OK, TX, LA, MS, AR, MO, IL, IN, KY, TN, GA, AL, FL, SC}},
 {10 824.2, {AL, LA, MS, AR, OK, TX, NM, AZ, NV, CA, OR, WA, ID, MT, UT, CO,
    WY, ND, SD, NE, KS, MO, IL, IA, MN, WI, MI, IN, TN, KY, OH, WV,
    PA, NY, VT, ME, NH, MA, RI, CT, NJ, DE, MD, VA, NC, SC, GA, FL}}}
```

{{10 834.1,  }, {12 860.9,  },

{13 998.2,  }, {10 648.5,  },

{10 894.9,  }, {11 075.7,  },

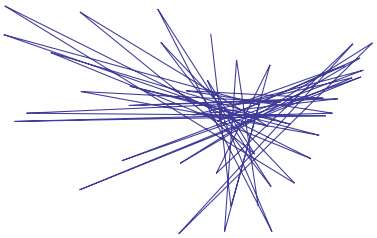
{10 834.1,  }, {46 865.4,  },

{10 648.5,  }, {10 824.2,  }}

LONGEST TOURS

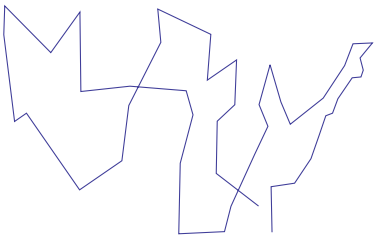
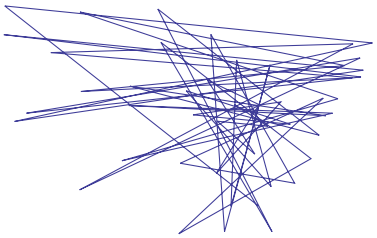
Mathematica found with no method specified:

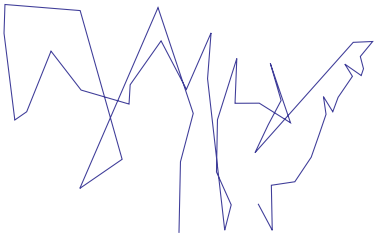
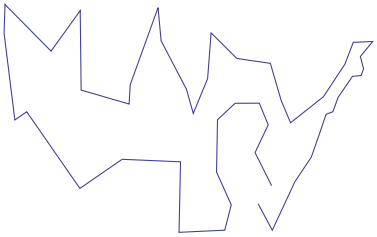
```
{{70 522.1, {AL, WI, LA, IN, OR, KY, MT, TN, IA, FL, ND, GA,
   SD, SC, WA, NC, ID, WV, WY, VA, UT, OH, NV, MD, CA, DE, NE, PA, CO, NJ,
   KS, CT, AZ, RI, NM, MA, IL, NY, MO, NH, OK, ME, TX, VT, AR, MI, MS, MN}}}
```

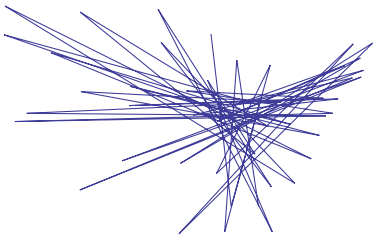
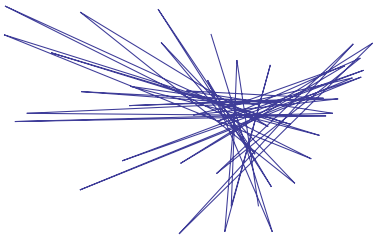
{{70 522.1,  }}

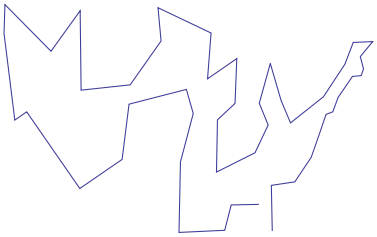
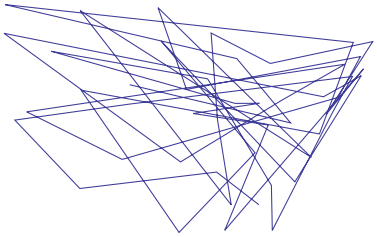
Trying ten different methods:

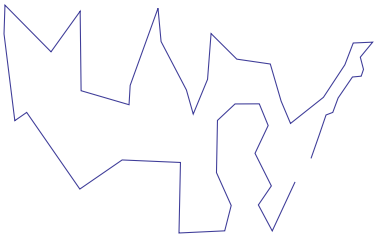
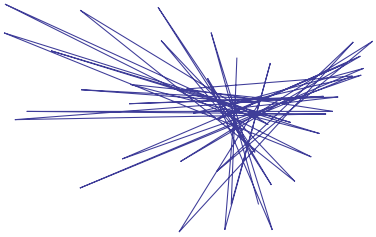
```
{{11 174.4, {AL, AR, MO, IL, WI, IA, MN, ND, SD, CO, NM, AZ,
    NV, CA, OR, WA, ID, MT, UT, WY, NE, KS, OK, TX, LA, MS, TN, KY, IN, MI,
    OH, WV, PA, NY, VT, ME, NH, MA, RI, CT, NJ, DE, MD, VA, NC, SC, GA, FL}},
 {68 115.3, {AL, WA, ME, CA, MA, OR, RI, NV, NH, AZ, VT, ID, CT, UT, NY,
    MT, NJ, NM, DE, CO, MD, WY, VA, ND, FL, SD, NC, TX, PA, OK, SC, MN,
    LA, MI, MS, WI, GA, NE, WV, KS, OH, AR, IN, IA, TN, MO, KY, IL}},
 {13 998.2, {AL, FL, GA, SC, NC, VA, MD, PA, DE, NJ, CT, NY, RI, MA, NH, ME,
    VT, TN, KY, OH, MI, WV, IN, IL, WI, MO, AR, MS, LA, IA, MN, NE,
    SD, WY, CO, UT, ID, NV, CA, OR, WA, MT, NM, AZ, ND, KS, OK, TX}},
 {10 648.5, {AL, FL, SC, NC, VA, MD, DE, NJ, CT, RI, MA, NH, ME, VT, NY, PA,
    WV, OH, MI, WI, MN, IA, KS, NE, SD, ND, WY, CO, UT, MT, ID, WA,
    OR, CA, NV, AZ, NM, OK, TX, LA, MS, AR, MO, IL, IN, KY, TN, GA}},
 {70 522.1, {AL, WI, LA, IN, OR, KY, MT, TN, IA, FL, ND, GA, SD, SC, WA, NC,
    ID, WV, WY, VA, UT, OH, NV, MD, CA, DE, NE, PA, CO, NJ, KS, CT,
    AZ, RI, NM, MA, IL, NY, MO, NH, OK, ME, TX, VT, AR, MI, MS, MN}},
 {70 587.1, {AL, WI, MS, MI, LA, IL, NY, OK, ME, TX, VT, AR, NH, MO, MA, NM,
    RI, AZ, CT, KS, NJ, CO, PA, NE, DE, CA, MD, NV, OH, UT, IN, WY,
    VA, ID, WV, OR, NC, WA, KY, MT, SC, SD, TN, ND, GA, IA, FL, MN}},
 {10 955.5, {AL, MS, LA, TX, OK, KS, NE, CO, NM, AZ, NV, CA, OR, WA, ID, MT,
    UT, WY, SD, ND, MN, IA, WI, IL, MO, AR, TN, KY, IN, MI, OH, WV,
    PA, NY, VT, ME, NH, MA, RI, CT, NJ, DE, MD, VA, NC, SC, GA, FL}},
 {46 865.4, {WY, WV, WI, WA, VT, VA, UT, TX, TN, SD, SC, RI, PA, OR, OK, OH,
    NY, NV, NM, NJ, NH, NE, ND, NC, MT, MS, MO, MN, MI, ME, MD, MA,
    LA, KY, KS, IN, IL, ID, IA, GA, FL, DE, CT, CO, CA, AZ, AR, AL}},
 {10 648.5, {NC, VA, MD, DE, NJ, CT, RI, MA, NH, ME, VT, NY, PA, WV, OH, MI,
    WI, MN, IA, KS, NE, SD, ND, WY, CO, UT, MT, ID, WA, OR, CA, NV,
    AZ, NM, OK, TX, LA, MS, AR, MO, IL, IN, KY, TN, GA, AL, FL, SC}},
 {70 567.4, {AL, MN, FL, IA, GA, ND, TN, SD, SC, MT, KY, WA, NC, OR, WV, ID,
    VA, WY, OH, UT, IN, TX, VT, AR, ME, OK, NH, MO, NY, NM, RI, KS,
    NJ, CO, PA, CA, MD, NV, DE, NE, CT, AZ, MA, IL, LA, MI, MS, WI}}}
```

{{11 174.4,  }, {68 115.3,  },

{13 998.2,  }, {10 648.5,  },

{70 522.1,  }, {70 587.1,  },

{10 955.5,  }, {46 865.4,  },

{10 648.5,  }, {70 567.4,  }}

END OUTPUT

In[161]:=
```
(*
fn = "/home/mark/talk/mma/stem-web-resources/doc.kml";
Import[fn]
Import[fn, "Elements"]
Import[fn, {"Data", 1}]
t = Table[
  Import[fn, {"Data",1, "Geometry", i, 2}],
  {i,1, 50}
 ];
Graphics3D[t]
fn = "/home/mark/talk/mma/stem-web-resources/us_states.kml";
Import[fn]
Import[fn,"Elements"]
Import[fn,{"Data",1}]
Import[fn, "Graphics"]
t = Table[
  Import[fn, {"Data",1, "Geometry", i, 2}],
  {i,1, 50}
 ];
Graphics[t]
*)
```

In[162]:=
```
endtime = AbsoluteTime[];

Print["program took ", endtime - starttime, " seconds to run"];
```

```
program took 70.102629 seconds to run
```