In [1]:
```
]add LightGraphs
```

```
   Updating registry at `C:\Users\15145\.julia\registries\General`
  Resolving package versions...
No Changes to `C:\Users\15145\.julia\environments\v1.5\Project.toml`
No Changes to `C:\Users\15145\.julia\environments\v1.5\Manifest.toml`
```

In [2]:
```
]add SimpleWeightedGraphs
```

```
  Resolving package versions...
No Changes to `C:\Users\15145\.julia\environments\v1.5\Project.toml`
No Changes to `C:\Users\15145\.julia\environments\v1.5\Manifest.toml`
```

In [3]:
```
using LightGraphs, SimpleWeightedGraphs
```

In [4]:
```
sources = [1,   2,  3,  4,  5,  6,  6,  7,  8,  9,  9,  10, 11, 4,  12, 9,  8,  1
```

Out[4]:
```
420-element Array{Int64,1}:
  1
  2
  3
  4
  5
  6
  6
  7
  8
  9
  9
 10
 11
  ⋮
  7
 19
  6
 18
  2
  6
 20
 16
 14
  6
 15
 16
```

In [5]:
```
destinations = [2,  1,  4,  3,  6,  5,  7,  6,  9,  8,  10, 9,  4,  11, 9,  12, 1
```

Out[5]:
```
420-element Array{Int64,1}:
  2
  1
  4
  3
  6
  5
  7
  6
  9
  8
 10
  9
  4
  ⋮
 19
  7
 18
  6
  6
  2
 16
 20
  6
 14
 16
 15
```

In [6]:
```
weights = [1.95080273848232,    1.95080273848232,   1.93033824330216,   1.9303382
```

Out[6]:
```
420-element Array{Float64,1}:
 1.95080273848232
 1.95080273848232
 1.93033824330216
 1.93033824330216
 1.92789657770857
 1.92789657770857
 1.92054412603243
 1.92054412603243
 1.88234487757313
 1.88234487757313
 1.88192481223769
 1.88192481223769
 1.87407377370522
 ⋮
 0.0896730123303072
 0.0896730123303072
 0.0872900276029559
 0.0872900276029559
 0.0561307271138995
 0.0561307271138995
 0.0459526090476401
 0.0459526090476401
 0.0409802469157487
 0.0409802469157487
 0.0052192547240316
 0.0052192547240316
```

In [7]:
```
g = SimpleWeightedGraph(sources, destinations, weights)
```

Out[7]:
```
{21, 210} undirected simple Int64 graph with Float64 weights
```

In [8]:
```
] add GraphPlot
```

```
  Resolving package versions...
No Changes to `C:\Users\15145\.julia\environments\v1.5\Project.toml`
No Changes to `C:\Users\15145\.julia\environments\v1.5\Manifest.toml`
```

In [9]:
```
using GraphPlot
```

In [12]:
```
gplot(g)
```

Out[12]:

In [14]:
```
nodelabel = 1:nv(g)
gplot(g, nodelabel=nodelabel)
```

Out[14]: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

In [15]:
```
gplot(g, nodelabel=nodelabel, nodelabeldist=1.5, nodelabelangleoffset=π/4,nodefil
```

Out[15]: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

In [16]:
```
g_mst=kruskal_mst(g,minimize=false)
```

Out[16]:
```
20-element Array{SimpleWeightedEdge{Int64,Float64},1}:
 Edge 1 => 2 with weight 3.90160547696464
 Edge 3 => 4 with weight 3.86067648660432
 Edge 5 => 6 with weight 3.85579315541714
 Edge 6 => 7 with weight 3.84108825206486
 Edge 8 => 9 with weight 3.76468975514626
 Edge 9 => 10 with weight 3.76384962447538
 Edge 4 => 11 with weight 3.74814754741044
 Edge 9 => 12 with weight 3.74090333248784
 Edge 12 => 13 with weight 3.57838682985612
 Edge 5 => 8 with weight 3.40879430952012
 Edge 2 => 14 with weight 3.40489580255538
 Edge 3 => 10 with weight 3.33868416288256
 Edge 4 => 15 with weight 3.32608933095458
 Edge 14 => 16 with weight 3.27882562562348
 Edge 12 => 17 with weight 3.2571516743888
 Edge 1 => 3 with weight 3.2130693301372
 Edge 14 => 18 with weight 3.16512447195928
 Edge 18 => 19 with weight 2.93316914265982
 Edge 12 => 21 with weight 2.76064343688234
 Edge 17 => 20 with weight 2.69518548317692
```

In [17]:
```
g_mst = SimpleWeightedGraph(size(g)[1]) #Create a new graph
for ew in kruskal_mst(g)
   add_edge!(g_mst,ew.src,ew.dst,ew.weight)
end
gplot(g_mst, nodelabel=nodelabel, nodelabeldist=1.5, nodelabelangleoffset=π/4,noc
```

Out[17]: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

In [47]:
```
# nodes size proportional to their degree
nodesize = [LightGraphs.outdegree(g, v) for v in LightGraphs.vertices(g)]
gplot(g, nodesize=nodesize)
```

Out[47]:

In [23]:
```
import Pkg
Pkg.add("Colors")

using Colors

# Generate n maximally distinguishable colors in LCHab space.
nodefillc = distinguishable_colors(nv(g), colorant"blue")
gplot(g, nodefillc=nodefillc, nodelabel=nodelabel, nodelabeldist=1.8, nodelabelar
```

```
  Resolving package versions...
Updating `C:\Users\15145\.julia\environments\v1.5\Project.toml`
  [5ae59095] + Colors v0.12.6
No Changes to `C:\Users\15145\.julia\environments\v1.5\Manifest.toml`
```

Out[23]: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

In [24]:
```
alphas = nodesize/maximum(nodesize)
nodefillc = [RGBA(0.0,0.8,0.8,i) for i in alphas]
gplot(g, nodefillc=nodefillc)
```

Out[24]:

In [25]:
```
nodelabelsize = nodesize
gplot(g, nodelabelsize=nodelabelsize, nodesize=nodesize, nodelabel=nodelabel)
```

Out[25]: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

In [26]:
```
edgelabel = 1:LightGraphs.ne(g)
gplot(g, edgelabel=edgelabel, nodelabel=nodelabel)
```

Out[26]: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

In [27]:
```
edgelabel = 1:LightGraphs.ne(g)
gplot(g, edgelabel=edgelabel, nodelabel=nodelabel, edgelabeldistx=0.5, edgelabelc
```

Out[27]:
```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66
67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97
98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142
143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164
165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186
187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208
209 210 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
```

In [32]:
```
import Pkg
Pkg.add("Cairo")
import Pkg
Pkg.add("Compose")
using Cairo, Compose
draw(PDF("GBM.pdf", 16cm, 16cm), gplot(g))
```

```
  Resolving package versions...
No Changes to `C:\Users\15145\.julia\environments\v1.5\Project.toml`
No Changes to `C:\Users\15145\.julia\environments\v1.5\Manifest.toml`
  Resolving package versions...
Updating `C:\Users\15145\.julia\environments\v1.5\Project.toml`
  [a81c6b42] + Compose v0.9.2
No Changes to `C:\Users\15145\.julia\environments\v1.5\Manifest.toml`
```

In [38]:
```
nv(g)
```

Out[38]:
```
21
```

In [39]:
```
ne(g)
```

Out[39]:
```
210
```

In [40]:
```
has_self_loops(g)
```

Out[40]:
```
false
```

In [45]:
```
dfs_tree(g, 1)
```

Out[45]:
```
{21, 20} directed simple Int64 graph
```

In [48]:
```
diffusion(g, 0.9, 10)
```

Out[48]:
```
10-element Array{Array{Int64,1},1}:
 [1]
 [4, 2, 3, 16, 11, 5, 21, 20, 7, 9, 13, 10, 14, 19, 17, 8, 15, 6, 18]
 [12]
 []
 []
 []
 []
 []
 []
 []
```

In [49]:
```
is_connected(g)
```

Out[49]:
```
true
```

In [56]:
```
bridges(g)
```

Out[56]:
```
LightGraphs.SimpleGraphs.SimpleEdge{Int64}[]
```

In [63]:
```
kruskal_mst(g)
```

Out[63]:
```
20-element Array{SimpleWeightedEdge{Int64,Float64},1}:
 Edge 15 => 16 with weight 0.0104385094480632
 Edge 6 => 14 with weight 0.0819604938314974
 Edge 16 => 20 with weight 0.0919052180952802
 Edge 2 => 6 with weight 0.112261454227799
 Edge 6 => 18 with weight 0.1745800552059118
 Edge 7 => 19 with weight 0.1793460246606144
 Edge 7 => 18 with weight 0.214267540104052
 Edge 6 => 20 with weight 0.268741936727874
 Edge 14 => 17 with weight 0.280086255382158
 Edge 1 => 5 with weight 0.280333486221224
 Edge 13 => 16 with weight 0.290421376645734
 Edge 2 => 5 with weight 0.303657064139408
 Edge 7 => 21 with weight 0.440004408251422
 Edge 7 => 12 with weight 0.523314510162542
 Edge 11 => 16 with weight 0.54468813973318
 Edge 3 => 16 with weight 0.91404563619835
 Edge 10 => 16 with weight 1.079218573553308
 Edge 4 => 5 with weight 1.091783161767466
 Edge 9 => 16 with weight 1.277542869485316
 Edge 7 => 8 with weight 1.666436712226092
```

In [66]:
```
kruskal_mst(g, minimize= true)
```

Out[66]:
```
20-element Array{SimpleWeightedEdge{Int64,Float64},1}:
 Edge 15 => 16 with weight 0.0104385094480632
 Edge 6 => 14 with weight 0.0819604938314974
 Edge 16 => 20 with weight 0.0919052180952802
 Edge 2 => 6 with weight 0.112261454227799
 Edge 6 => 18 with weight 0.1745800552059118
 Edge 7 => 19 with weight 0.1793460246606144
 Edge 7 => 18 with weight 0.214267540104052
 Edge 6 => 20 with weight 0.268741936727874
 Edge 14 => 17 with weight 0.280086255382158
 Edge 1 => 5 with weight 0.280333486221224
 Edge 13 => 16 with weight 0.290421376645734
 Edge 2 => 5 with weight 0.303657064139408
 Edge 7 => 21 with weight 0.440004408251422
 Edge 7 => 12 with weight 0.523314510162542
 Edge 11 => 16 with weight 0.54468813973318
 Edge 3 => 16 with weight 0.91404563619835
 Edge 10 => 16 with weight 1.079218573553308
 Edge 4 => 5 with weight 1.091783161767466
 Edge 9 => 16 with weight 1.277542869485316
 Edge 7 => 8 with weight 1.666436712226092
```

In [18]:
```
prim_mst(g)
```

Out[18]:
```
20-element Array{LightGraphs.SimpleGraphs.SimpleEdge{Int64},1}:
 Edge 5 => 2
 Edge 16 => 3
 Edge 5 => 4
 Edge 1 => 5
 Edge 2 => 6
 Edge 18 => 7
 Edge 7 => 8
 Edge 16 => 9
 Edge 16 => 10
 Edge 16 => 11
 Edge 7 => 12
 Edge 16 => 13
 Edge 6 => 14
 Edge 16 => 15
 Edge 20 => 16
 Edge 14 => 17
 Edge 6 => 18
 Edge 7 => 19
 Edge 6 => 20
 Edge 7 => 21
```

In [79]:
```
dijkstra_shortest_paths(g, 1)
```

Out[79]:
```
LightGraphs.DijkstraState{Float64,Int64}([0, 5, 7, 5, 1, 1, 19, 7, 1, 1  …  7,
16, 6, 5, 15, 1, 6, 1, 6, 5], [0.0, 0.583990550360632, 1.5074266499527704, 1.37
211664798869, 0.280333486221224, 0.562370878960302, 0.5768629731356284, 2.24329
968536172, 1.51428416533707, 1.80511810035208  …  1.1001774832981703, 1.1835570
644024451, 0.6443313727917994, 0.882697178308648, 0.8931356877567113, 0.7476347
38708264, 0.7369509341662138, 0.397516948475014, 0.831112815688176, 1.009971848
648724], [Int64[], Int64[], Int64[], Int64[], Int64[], Int64[], Int64[], Int64
[], Int64[], Int64[]  …  Int64[], Int64[], Int64[], Int64[], Int64[], Int64[],
Int64[], Int64[], Int64[], Int64[]], [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,
1.0, 1.0  …  1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0], Int64[])
```

In [83]:
```
bellman_ford_shortest_paths(g, 3)
```

Out[83]:
```
LightGraphs.BellmanFordState{Float64,Int64}([19, 6, 0, 16, 15, 20, 3, 16, 16, 1
6  …  7, 16, 6, 16, 3, 19, 7, 7, 16, 7], [1.5074266499527704, 1.386954245249303
3, 0.0, 2.45950488485445, 1.5268478377338373, 1.2746927910215042, 0.93056367681
7142, 2.586787496882962, 2.191588505683666, 1.993264209751658  …  1.45387818697
96842, 1.204467012844084, 1.3566532848530017, 0.9244841456464132, 0.91404563619
835, 1.5654158538553664, 1.144831216921194, 1.1099097014777564, 1.0059508542936
302, 1.370568085068564])
```

In [87]:
```
center(g)
```

Out[87]:
```
1-element Array{Int64,1}:
 16
```

In [88]:
```
diameter(g)
```

Out[88]:
```
3.03653502017804
```

In [91]:
```
betweenness_centrality(g)
```

Out[91]:
```
21-element Array{Float64,1}:
 0.010526315789473684
 0.07368421052631578
 0.0
 0.0
 0.08421052631578947
 0.35789473684210527
 0.18947368421052632
 0.0
 0.0
 0.0
 0.0
 0.0
 0.0
 0.06315789473684211
 0.03684210526315789
 0.39473684210526316
 0.0
 0.1368421052631579
 0.042105263157894736
 0.21578947368421053
 0.0
```

In [92]:
```
closeness_centrality(g, normalize=true)
```

Out[92]:
```
21-element Array{Float64,1}:
 1.00568006810535
 1.28228332448121
 0.6810465548638197
 0.5399981819200532
 1.1111305974705536
 1.3986335488404606
 1.1622026549549214
 0.46598754346061144
 0.5433036587062456
 0.5644577256301071
 0.8287742851753542
 0.7570580427717716
 0.9669506905748508
 1.3101145574275725
 1.305299890225991
 1.3187765188083798
 1.012381575950885
 1.2553508079147437
 1.2076562708175007
 1.3131780698712494
 0.8168653425104621
```

In [93]:
```
degree_centrality(g)
```

Out[93]:
```
21-element Array{Float64,1}:
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
```

In [94]:
```
indegree_centrality(g)
```

Out[94]:
```
21-element Array{Float64,1}:
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
```

In [96]:
```
eigenvector_centrality(g)
```

Out[96]:
```
21-element Array{Float64,1}:
 0.20952492054502686
 0.19280566699438395
 0.2602538323549946
 0.26255840351454435
 0.18342525649480623
 0.1600103914752922
 0.15050518647907987
 0.27964830316574363
 0.2679670167451129
 0.27295078094286584
 0.2496014861912524
 0.2616071859804524
 0.25287534535993383
 0.19844549721840102
 0.2144802502740931
 0.13263948522720445
 0.195197927938822
 0.1827520195240194
 0.17009477111255764
 0.1960959476075155
 0.19823610334943964
```

In [100]:
```
pagerank(g)
```

Out[100]:
```
21-element Array{Float64,1}:
 0.04761904761904763
 0.04761904761904763
 0.04761904761904763
 0.04761904761904763
 0.04761904761904763
 0.04761904761904763
 0.04761904761904763
 0.04761904761904763
 0.04761904761904763
 0.04761904761904763
 0.04761904761904763
 0.04761904761904763
 0.04761904761904763
 0.04761904761904763
 0.04761904761904763
 0.04761904761904763
 0.04761904761904763
 0.04761904761904763
 0.04761904761904763
 0.04761904761904763
 0.04761904761904763
```

In [103]:
```
maximal_cliques(g)
```

Out[103]:
```
1-element Array{Array{Int64,1},1}:
 [1, 2, 16, 11, 21, 7, 9, 10, 19, 17  …  20, 6, 4, 13, 14, 3, 5, 15, 12, 18]
```

In [104]:
```
global_clustering_coefficient(g)
```

Out[104]:
```
1.0
```

In [112]:
```
core_periphery_deg(g)
```

Out[112]:
```
21-element Array{Int64,1}:
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 2
```

In [19]:
```
local_clustering_coefficient(g,1)
```

Out[19]:
```
1.0
```

In [20]:
```
local_clustering_coefficient(g,2)
```

Out[20]:
```
1.0
```