

```

1  local function optimize(egs,      cluster,leaves,row1,row2)
2      cluster = CLUSTER(egs)
3      local function order(a,b) return a.egs:betters(b.egs) end
4      for rank,leaf in pairs(quicksort(cluster:leaves(), order)) do
5          leaf.rank = rank end
6      return cluster end
7
8  function CLUSTER.project(i,row)
9      return cosine(i.top:dist(row, i.left), i.top:dist(row, i.right), i.c) end
10
11 function CLUSTER.where(i,row)
12     if i:leaf() then return i end
13     if i:project(row) <= i.border
14     then return i.lefts and i.lefts:where( row) or i
15     else return i.rights and i.rights:where( row) or i end end
16
17 function CLUSTER.better(i,row1,row2,      where1, where2)
18     where1, where2 = i:where(row1), i:where(row2)
19     if where1.rank > where2.rank then return false
20     elseif where1.rank < where2.rank then return true
21     else return where1:xbetter(row1,row2) end end
22
23 function CLUSTER.xbetter(i,row1,row2, x1,x2)
24     x1,x2 = i:project(row1), i:project(row2)
25     return i.egs:better(i.left, i.right) and x1 <= x2 or x1 > x2 end
26
27 function CLUSTER.leaves(i, out)
28     out = out or {}
29     if i:leaf() then push(out,i) end
30     if i.lefts then i.lefts:leaves(out) end
31     if i.rights then i.rights:leaves(out) end
32     return out end
33
34 local function optimize(egs,      cluster,leaves,row1,row2)
35 local function optimize(egs,      cluster,leaves,row1,row2)
36 local function optimize(egs,      cluster,leaves,row1,row2)
37     cluster = CLUSTER(egs)
38     local function order(a,b) return a.egs:betters(b.egs) end
39     for rank,leaf in pairs(quicksort(cluster:leaves(), order)) do
40         leaf.rank = rank end
41     return cluster end
42
43 function CLUSTER.project(i,row)
44     return cosine(i.top:dist(row, i.left), i.top:dist(row, i.right), i.c) end
45
46 function CLUSTER.where(i,row)
47     if i:leaf() then return i end
48     if i:project(row) <= i.border
49     then return i.lefts and i.lefts:where( row) or i
50     else return i.rights and i.rights:where( row) or i end end
51
52 function CLUSTER.better(i,row1,row2,      where1, where2)
53     where1, where2 = i:where(row1), i:where(row2)
54     if where1.rank > where2.rank then return false
55     elseif where1.rank < where2.rank then return true
56     else return where1:xbetter(row1,row2) end end
57
58 function CLUSTER.xbetter(i,row1,row2, x1,x2)
59     x1,x2 = i:project(row1), i:project(row2)
60     return i.egs:better(i.left, i.right) and x1 <= x2 or x1 > x2 end
61
62 function CLUSTER.leaves(i, out)
63     out = out or {}
64     if i:leaf() then push(out,i) end
65     if i.lefts then i.lefts:leaves(out) end
66     if i.rights then i.rights:leaves(out) end
67     return out end
68
69 cluster = CLUSTER(egs)
70 local function order(a,b) return a.egs:betters(b.egs) end
71 for rank,leaf in pairs(quicksort(cluster:leaves(), order)) do
72     leaf.rank = rank end
73 return cluster end
74
75 function CLUSTER.project(i,row)
76     return cosine(i.top:dist(row, i.left), i.top:dist(row, i.right), i.c) end
77
78 function CLUSTER.where(i,row)
79     if i:leaf() then return i end
80     if i:project(row) <= i.border
81     then return i.lefts and i.lefts:where( row) or i
82     else return i.rights and i.rights:where( row) or i end end
83
84 function CLUSTER.better(i,row1,row2,      where1, where2)
85     where1, where2 = i:where(row1), i:where(row2)
86     if where1.rank > where2.rank then return false
87     elseif where1.rank < where2.rank then return true
88     else return where1:xbetter(row1,row2) end end
89
90 function CLUSTER.xbetter(i,row1,row2, x1,x2)
91     x1,x2 = i:project(row1), i:project(row2)
92     return i.egs:better(i.left, i.right) and x1 <= x2 or x1 > x2 end
93
94 function CLUSTER.leaves(i, out)
95     out = out or {}
96     if i:leaf() then push(out,i) end
97     if i.lefts then i.lefts:leaves(out) end
98     if i.rights then i.rights:leaves(out) end
99     return out end
100
101 local function optimize(egs,      cluster,leaves,row1,row2)
102     cluster = CLUSTER(egs)
103     local function order(a,b) return a.egs:betters(b.egs) end
104     for rank,leaf in pairs(quicksort(cluster:leaves(), order)) do
105         leaf.rank = rank end
106     return cluster end
107
108 function CLUSTER.project(i,row)
109     return cosine(i.top:dist(row, i.left), i.top:dist(row, i.right), i.c) end
110
111 function CLUSTER.where(i,row)
112     if i:leaf() then return i end
113     if i:project(row) <= i.border
114     then return i.lefts and i.lefts:where( row) or i
115     else return i.rights and i.rights:where( row) or i end end
116
117 function CLUSTER.better(i,row1,row2,      where1, where2)
118     where1, where2 = i:where(row1), i:where(row2)
119     if where1.rank > where2.rank then return false
120     elseif where1.rank < where2.rank then return true
121     else return where1:xbetter(row1,row2) end end
122
123 function CLUSTER.xbetter(i,row1,row2, x1,x2)
124     x1,x2 = i:project(row1), i:project(row2)
125     return i.egs:better(i.left, i.right) and x1 <= x2 or x1 > x2 end
126
127 function CLUSTER.leaves(i, out)
128     out = out or {}
129     if i:leaf() then push(out,i) end
130     if i.lefts then i.lefts:leaves(out) end
131     if i.rights then i.rights:leaves(out) end
132     return out end
133
134 luster = CLUSTER(egs)
135 local function order(a,b) return a.egs:betters(b.egs) end
136 for rank,leaf in pairs(quicksort(cluster:leaves(), order)) do
137     leaf.rank = rank end
138 return cluster end
139
140 function CLUSTER.project(i,row)
141     return cosine(i.top:dist(row, i.left), i.top:dist(row, i.right), i.c) end
142
143 function CLUSTER.where(i,row)
144     if i:leaf() then return i end
145     if i:project(row) <= i.border
146     then return i.lefts and i.lefts:where( row) or i
147     else return i.rights and i.rights:where( row) or i end end
148
149 function CLUSTER.better(i,row1,row2,      where1, where2)
150     where1, where2 = i:where(row1), i:where(row2)
151     if where1.rank > where2.rank then return false
152     elseif where1.rank < where2.rank then return true
153     else return where1:xbetter(row1,row2) end end
154
155 function CLUSTER.xbetter(i,row1,row2, x1,x2)
156     x1,x2 = i:project(row1), i:project(row2)
157     return i.egs:better(i.left, i.right) and x1 <= x2 or x1 > x2 end
158
159 function CLUSTER.leaves(i, out)
160     out = out or {}
161     if i:leaf() then push(out,i) end
162     if i.lefts then i.lefts:leaves(out) end
163     if i.rights then i.rights:leaves(out) end
164     return out end
165
166

```