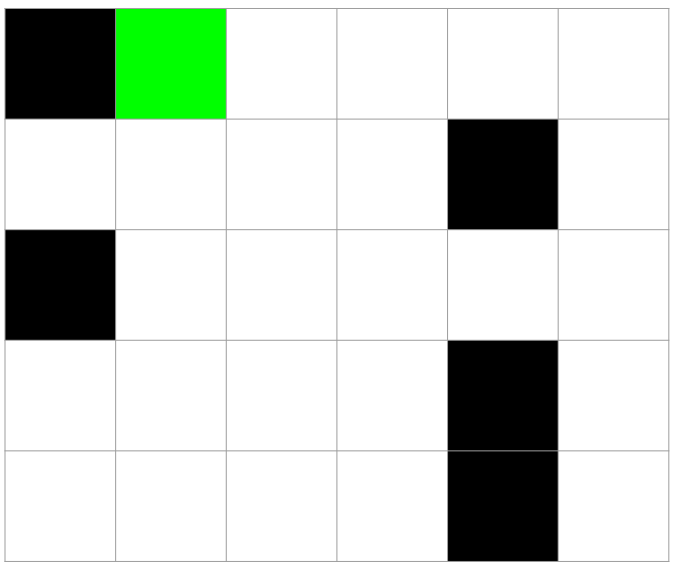
MARKOV DECISION TREES AND NEURAL NETWORKS; ATARI GAME;
BLACK ARE OBSTACLES AND GREEN HAS TO MOVE DOWN DEVIATING FROM BLACK OBSTACLES AND FIND THE SHORTEST PATH; GREEN STARTS AT POSITION 2, FIRST LINE TIME T = 1 AND SO FORTH;

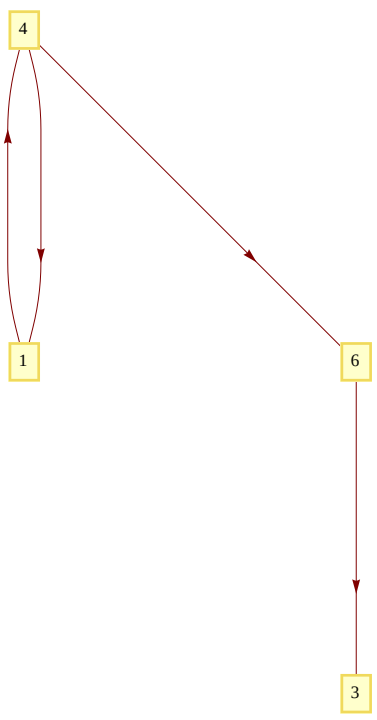
In[1]:=
```
a = Table[Insert[Table[0, {5}], 1, RandomInteger[{1, 5}]], {5}]
```
Out[1]=
```
{{1, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 1, 0}, {1, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 1, 0}, {0, 0, 0, 0, 1, 0}}
```
In[2]:=
```
c1 = {Delete[Insert[a[[1]], 2, 2], 3], a[[2]], a[[3]], a[[4]], a[[5]]}
```
Out[2]=
```
{{1, 2, 0, 0, 0, 0}, {0, 0, 0, 0, 1, 0}, {1, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 1, 0}, {0, 0, 0, 0, 1, 0}}
```
In[55]:=
```
ArrayPlot[c1, ColorRules → {0 → White, 1 → Black, 2 → Green}, Mesh → True]
```
Out[55]=



In[4]:=
```
b = Partition[Flatten[Position[a[[#]], 0] & /@ Table[k, {k, 1, Dimensions[a][[1]], 1}]], 5]
```
Out[4]=
```
{{2, 3, 4, 5, 6}, {1, 2, 3, 4, 6}, {2, 3, 4, 5, 6}, {1, 2, 3, 4, 6}, {1, 2, 3, 4, 6}}
```

MARKOV CHAINS : RANDOM EXAMPLE

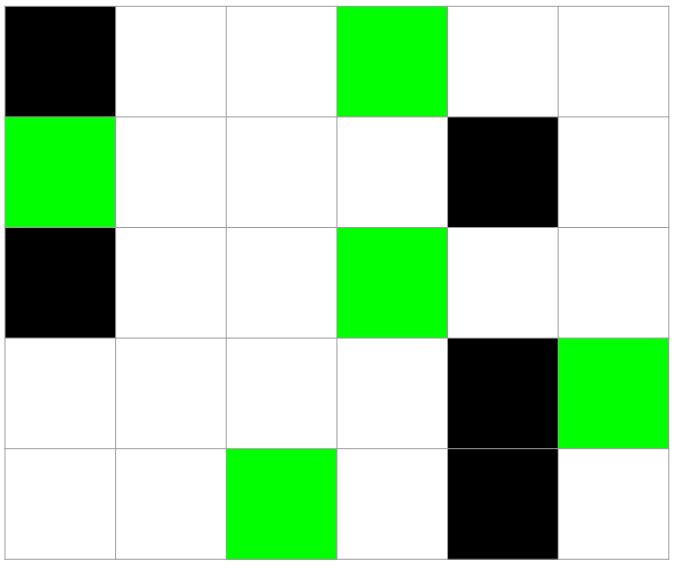
In[6]:=
```
h = b[[#]][[RandomInteger[{1, 5}]]] & /@ Table[k, {k, 1, Dimensions[a][[1]], 1}]
```
Out[6]=
```
{4, 1, 4, 6, 3}
```
In[7]:=
```
g1 = {h[[1]] → h[[2]], h[[2]] → h[[3]], h[[3]] → h[[4]], h[[4]] → h[[5]]}
```
Out[7]=
```
{4 → 1, 1 → 4, 4 → 6, 6 → 3}
```
In[8]:=
```
LayeredGraphPlot[g1, VertexLabeling → True]
```
Out[8]=



In[10]:=
```
hh = Insert[a[[#]], 2, h[[#]]] & /@ Table[k, {k, 1, Dimensions[a][[1]], 1}]
```
Out[10]=
```
{{1, 0, 0, 2, 0, 0, 0}, {2, 0, 0, 0, 0, 1, 0}, {1, 0, 0, 2, 0, 0, 0}, {0, 0, 0, 0, 1, 2, 0}, {0, 0, 2, 0, 0, 1, 0}}
```
In[11]:=
```
f1 = Delete[hh[[#]], h[[#]] + 1] & /@ Table[k, {k, 1, Dimensions[a][[1]], 1}]
```
Out[11]=
```
{{1, 0, 0, 2, 0, 0}, {2, 0, 0, 0, 1, 0}, {1, 0, 0, 2, 0, 0}, {0, 0, 0, 0, 1, 2}, {0, 0, 2, 0, 1, 0}}
```

In[12]:=
```
ArrayPlot[f1, Mesh → True, ColorRules → {0 → White, 1 → Black, 2 → Green}]
```

Out[12]=



EXAMPLES OF PATHS STARTING FROM POSITION 2 IN ALL OPTIONS

In[26]:=
```
b
```

Out[26]=
```
{{2, 3, 4, 5, 6}, {1, 2, 3, 4, 6}, {2, 3, 4, 5, 6}, {1, 2, 3, 4, 6}, {1, 2, 3, 4, 6}}
```

In[29]:=
```
r = {b[[1]][[1]], b[[2]][[#]], b[[3]][[#]], b[[4]][[#]], b[[5]][[#]]} & /@ Table[k, {k, 1, Dimensions[b][[1]], 1}]
```

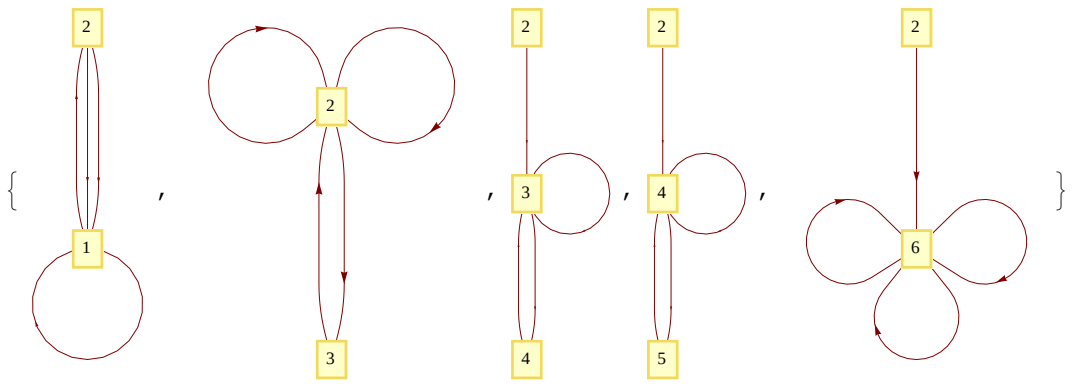
Out[29]=
```
{{2, 1, 2, 1, 1}, {2, 2, 3, 2, 2}, {2, 3, 4, 3, 3}, {2, 4, 5, 4, 4}, {2, 6, 6, 6, 6}}
```

In[30]:=
```
g12 = {r[[#]][[1]] → r[[#]][[2]], r[[#]][[2]] → r[[#]][[3]], r[[#]][[3]] → r[[#]][[4]], r[[#]][[4]] → r[[#]][[5]]} & /@
  Table[k, {k, 1, Dimensions[b][[1]], 1}]
```

Out[30]=
```
{{2 → 1, 1 → 2, 2 → 1, 1 → 1}, {2 → 2, 2 → 3, 3 → 2, 2 → 2}, {2 → 3, 3 → 4, 4 → 3, 3 → 3}, {2 → 4, 4 → 5, 5 → 4, 4 → 4}, {2 → 6, 6 → 6, 6 → 6, 6 → 6}}
```

In[52]:=
```
q = LayeredGraphPlot[g12[[#]], VertexLabeling → True] & /@ Table[k, {k, 1, Dimensions[b][[1]], 1}]
```

Out[52]=



In[33]:=
```
ft[t_] := Insert[a[[#]], 2, r[[t]][[#]]] & /@ Table[k, {k, 1, Dimensions[a][[1]], 1}]
```

In[35]:=
```
hh2 = ft /@ Table[k, {k, 1, Dimensions[r][[1]], 1}]
```

Out[35]=
```
{{{1, 2, 0, 0, 0, 0, 0}, {2, 0, 0, 0, 0, 1, 0}, {1, 2, 0, 0, 0, 0, 0}, {2, 0, 0, 0, 0, 1, 0}, {2, 0, 0, 0, 0, 1, 0}},
 {{1, 2, 0, 0, 0, 0, 0}, {0, 2, 0, 0, 0, 1, 0}, {1, 0, 2, 0, 0, 0, 0}, {0, 2, 0, 0, 0, 1, 0}, {0, 2, 0, 0, 0, 1, 0}},
 {{1, 2, 0, 0, 0, 0, 0}, {0, 0, 2, 0, 0, 1, 0}, {1, 0, 0, 2, 0, 0, 0}, {0, 0, 2, 0, 0, 1, 0}, {0, 0, 2, 0, 0, 1, 0}},
 {{1, 2, 0, 0, 0, 0, 0}, {0, 0, 0, 2, 0, 1, 0}, {1, 0, 0, 0, 2, 0, 0}, {0, 0, 0, 2, 0, 1, 0}, {0, 0, 0, 2, 0, 1, 0}},
 {{1, 2, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 1, 2, 0}, {1, 0, 0, 0, 0, 2, 0}, {0, 0, 0, 0, 1, 2, 0}, {0, 0, 0, 0, 1, 2, 0}}}
```

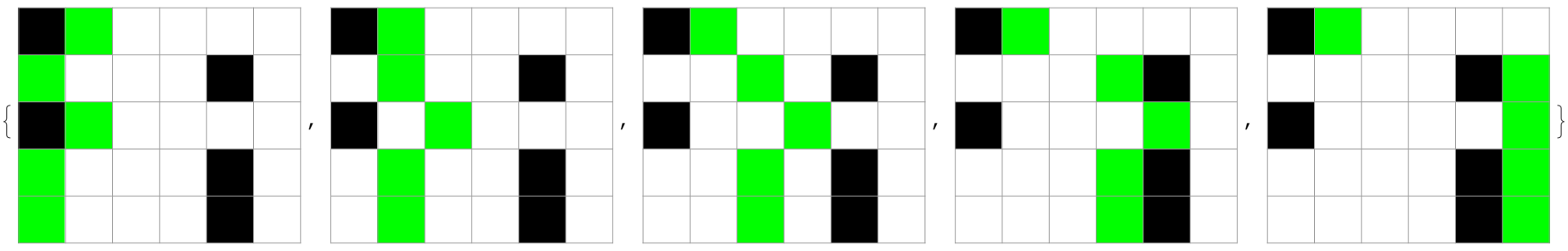
In[39]:=
```
gh[ty_] := Delete[hh2[[ty]][[#]], r[[ty]][[#]] + 1] & /@ Table[k, {k, 1, Dimensions[a][[1]], 1}]
```

In[41]:=
```
f13 = gh /@ Table[k, {k, 1, Dimensions[hh2][[1]], 1}]
```

Out[41]=
```
{{{1, 2, 0, 0, 0, 0}, {2, 0, 0, 0, 1, 0}, {1, 2, 0, 0, 0, 0}, {2, 0, 0, 0, 1, 0}, {2, 0, 0, 0, 1, 0}},
 {{1, 2, 0, 0, 0, 0}, {0, 2, 0, 0, 1, 0}, {1, 0, 2, 0, 0, 0}, {0, 2, 0, 0, 1, 0}, {0, 2, 0, 0, 1, 0}},
 {{1, 2, 0, 0, 0, 0}, {0, 0, 2, 0, 1, 0}, {1, 0, 0, 2, 0, 0}, {0, 0, 2, 0, 1, 0}, {0, 0, 2, 0, 1, 0}},
 {{1, 2, 0, 0, 0, 0}, {0, 0, 0, 2, 1, 0}, {1, 0, 0, 0, 2, 0}, {0, 0, 0, 2, 1, 0}, {0, 0, 0, 2, 1, 0}},
 {{1, 2, 0, 0, 0, 0}, {0, 0, 0, 0, 1, 2}, {1, 0, 0, 0, 0, 2}, {0, 0, 0, 0, 1, 2}, {0, 0, 0, 0, 1, 2}}}
```

In[51]:=
```
p = ArrayPlot[f13[[#]], Mesh → True, ColorRules → {0 → White, 1 → Black, 2 → Green}] & /@ Table[k, {k, 1, Dimensions[a][[1]], 1}]
```

Out[51]=



MOST EFFICIENT PATH (SHORTEST);
NEURAL NETWORK AS AN OPTIMIZATION ALGORITHM : THE ALGORITHM GOES BACK ON MARKOV CHAINS
POSSIBILITITES AND CHOOSES THE SHORTEST PATH BASED ON SMALLEST DISPLACEMENT OF GREEN SQUARE

In[43]:=
```
r
```

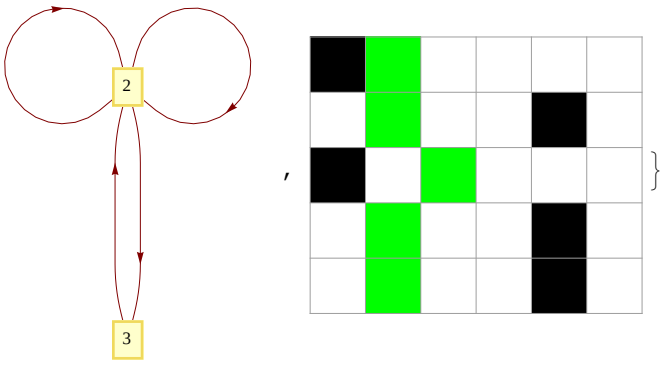
Out[43]=
```
{{2, 1, 2, 1, 1}, {2, 2, 3, 2, 2}, {2, 3, 4, 3, 3}, {2, 4, 5, 4, 4}, {2, 6, 6, 6, 6}}
```

In[48]:=
```
dg = Total[Abs[{r[[#]][[1]] - r[[#]][[2]], r[[#]][[2]] - r[[#]][[3]], r[[#]][[3]] - r[[#]][[4]], r[[#]][[4]] - r[[#]][[5]]}]] & /@
  Table[k, {k, 1, Dimensions[r][[1]], 1}]
```

Out[48]=
```
{3, 2, 3, 4, 4}
```

In[54]:=
```
{q[[Flatten[Position[dg, Min[dg]]][[1]]]], p[[Flatten[Position[dg, Min[dg]]][[1]]]]}
```

Out[54]=



,