CISC889 HW2 Shiyi Chen

Part 1: README

How to run?

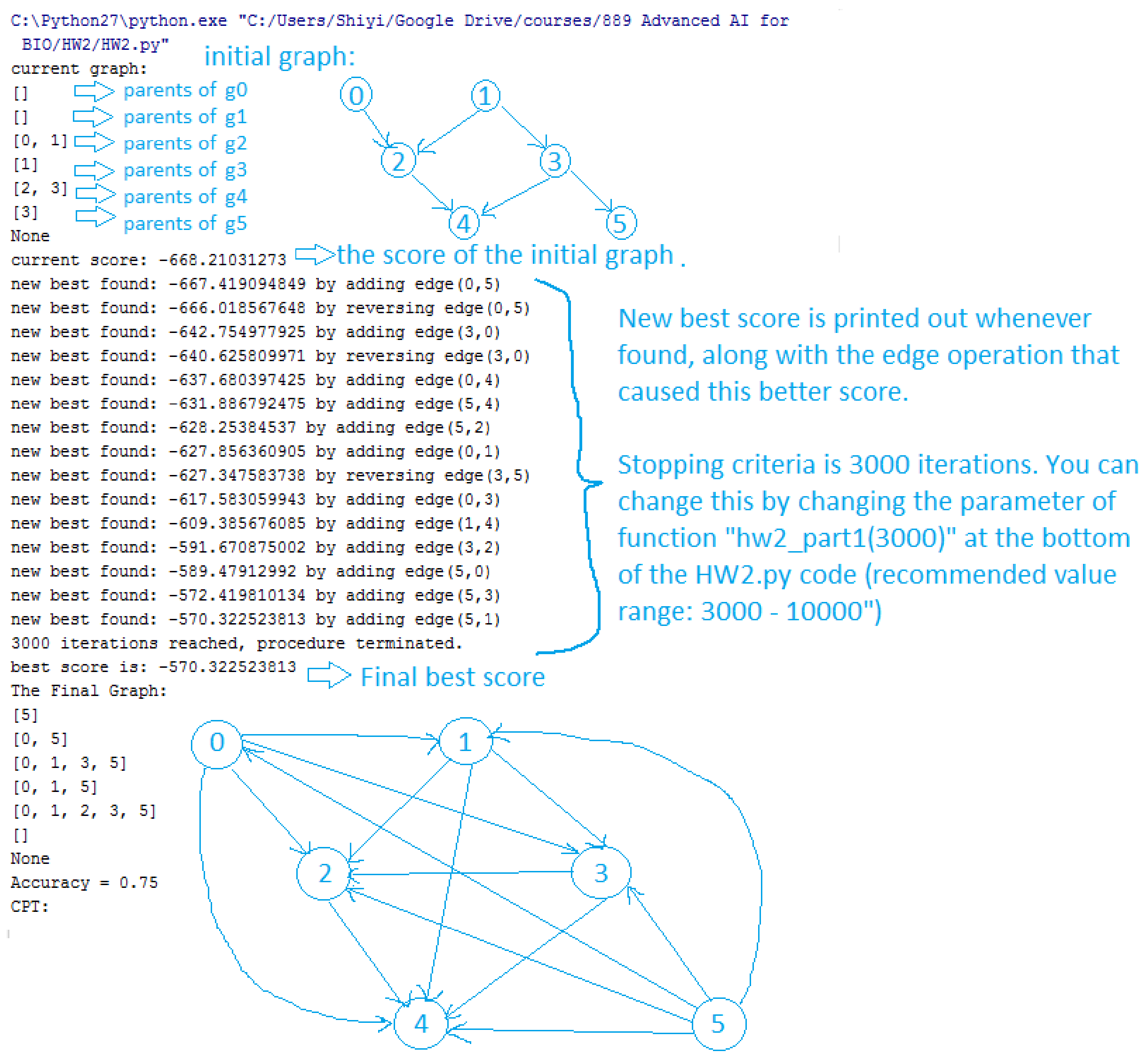
1. Unzip HW2.zip

2. At the top of HW2.py, there are “training_data_path” and “testing_data_path”. Change them to your own path.

3. Only compatible with Python 2.7

How to interpret the output?

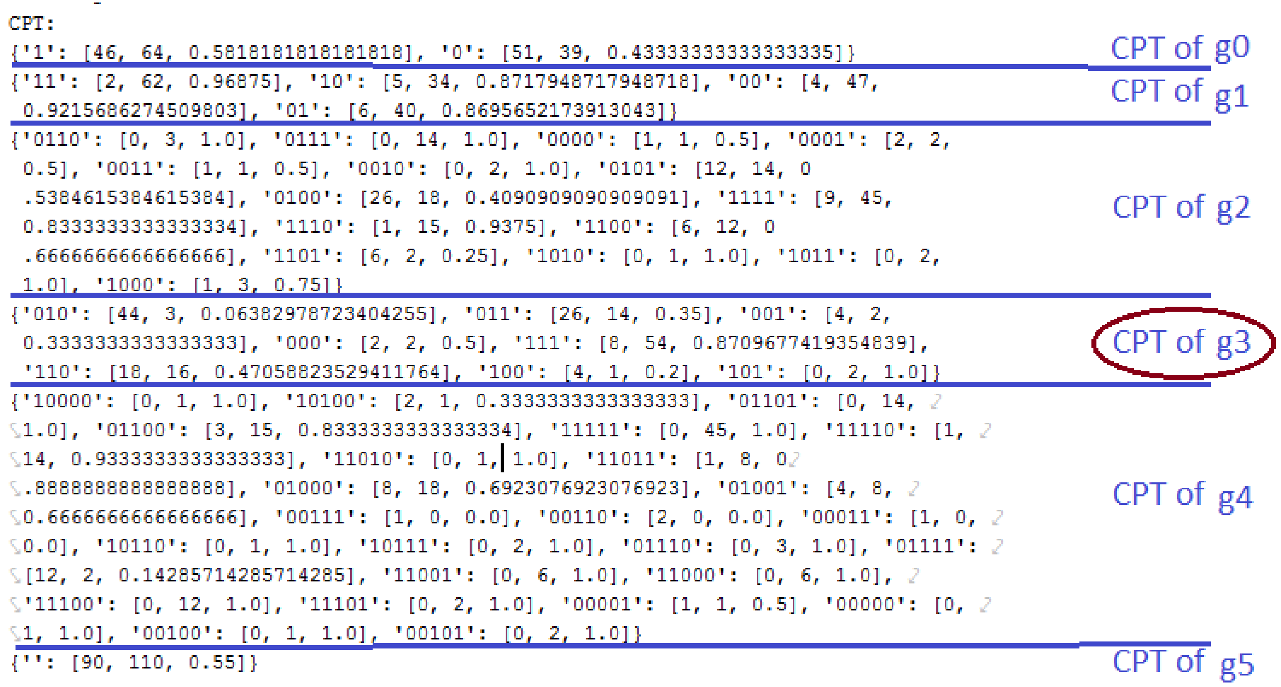
Here is an example output:



```
C:\Python27\python.exe "C:/Users/Shiyi/Google Drive/courses/889 Advanced AI for
 BIO/HW2/HW2.py"
current graph:
[]
[]
[0, 1]
[1]
[2, 3]
[3]
None
current score: -668.21031273
new best found: -667.419094849 by adding edge(0,5)
new best found: -666.018567648 by reversing edge(0,5)
new best found: -642.754977925 by adding edge(3,0)
new best found: -640.625809971 by reversing edge(3,0)
new best found: -637.680397425 by adding edge(0,4)
new best found: -631.886792475 by adding edge(5,4)
new best found: -628.25384537 by adding edge(5,2)
new best found: -627.856360905 by adding edge(0,1)
new best found: -627.347583738 by reversing edge(3,5)
new best found: -617.583059943 by adding edge(0,3)
new best found: -609.385676085 by adding edge(1,4)
new best found: -591.670875002 by adding edge(3,2)
new best found: -589.47912992 by adding edge(5,0)
new best found: -572.419810134 by adding edge(5,3)
new best found: -570.322523813 by adding edge(5,1)
3000 iterations reached, procedure terminated.
best score is: -570.322523813
The Final Graph:
[5]
[0, 5]
[0, 1, 3, 5]
[0, 1, 5]
[0, 1, 2, 3, 5]
[]
None
Accuracy = 0.75
CPT:
```

New best score is printed out whenever found, along with the edge operation that caused this better score.

Stopping criteria is 3000 iterations. You can change this by changing the parameter of function "hw2_part1(3000)" at the bottom of the HW2.py code (recommended value range: 3000 - 10000")



```
CPT:
{'1': [46, 64, 0.5818181818181818], '0': [51, 39, 0.43333333333333335]}
{'11': [2, 62, 0.96875], '10': [5, 34, 0.8717948717948718], '00': [4, 47,
 0.9215686274509803], '01': [6, 40, 0.8695652173913043]}
{'0110': [0, 3, 1.0], '0111': [0, 14, 1.0], '0000': [1, 1, 0.5], '0001': [2, 2,
 0.5], '0011': [1, 1, 0.5], '0010': [0, 2, 1.0], '0101': [12, 14, 0
 .5384615384615384], '0100': [26, 18, 0.4090909090909091], '1111': [9, 45,
 0.8333333333333334], '1110': [1, 15, 0.9375], '1100': [6, 12, 0
 .6666666666666666], '1101': [6, 2, 0.25], '1010': [0, 1, 1.0], '1011': [0, 2,
 1.0], '1000': [1, 3, 0.75]}
{'010': [44, 3, 0.06382978723404255], '011': [26, 14, 0.35], '001': [4, 2,
 0.3333333333333333], '000': [2, 2, 0.5], '111': [8, 54, 0.8709677419354839],
 '110': [18, 16, 0.47058823529411764], '100': [4, 1, 0.2], '101': [0, 2, 1.0]}
{'10000': [0, 1, 1.0], '10100': [2, 1, 0.3333333333333333], '01101': [0, 14,
1.0], '01100': [3, 15, 0.8333333333333334], '11111': [0, 45, 1.0], '11110': [1,
14, 0.9333333333333333], '11010': [0, 1, 1.0], '11011': [1, 8, 0
.8888888888888888], '01000': [8, 18, 0.6923076923076923], '01001': [4, 8,
0.6666666666666666], '00111': [1, 0, 0.0], '00110': [2, 0, 0.0], '00011': [1, 0,
0.0], '10110': [0, 1, 1.0], '10111': [0, 2, 1.0], '01110': [0, 3, 1.0], '01111':
[12, 2, 0.14285714285714285], '11001': [0, 6, 1.0], '11000': [0, 6, 1.0],
'11100': [0, 12, 1.0], '11101': [0, 2, 1.0], '00001': [1, 1, 0.5], '00000': [0,
1, 1.0], '00100': [0, 1, 1.0], '00101': [0, 2, 1.0]}
{'': [90, 110, 0.55]}
```

How to interpret this output of CPT? It has to be interpreted with the graph structure. Take “CPT of g3” as an example. Based on the graph, it has three parents: g0, g1, g5, so *‘010’: [44, 3, 0.0638]* means when g0 = 0, g1 = 1, g5 = 0, the number of times that g3 = 0 is 44, the number of times that g3 = 1 is 3, the probability of g3 = 1 is 0.0638. Overall, the CPT of g3 will look like this:

g0	g1	g5	P(g3) = 1
0	0	0	0.5
0	0	1	0.3333
0	1	0	0.0638
0	1	1	0.35
1	0	0	0.2
1	0	1	1.0
1	1	0	0.4705
1	1	1	0.8709

I need to point out that the CPT printed out by the program does not include a special case. For instance, g4 has five parents, so the CPT will consists of 32 lines. But some value combinations of its parents are not contained in the training data (eg: 10101). If we were to print it out, it would look like: *‘10101’: [0, 0, 0]*. Since the CPT output by this program does not contain all these combinations, it only has 25 cases, instead of 32 cases.

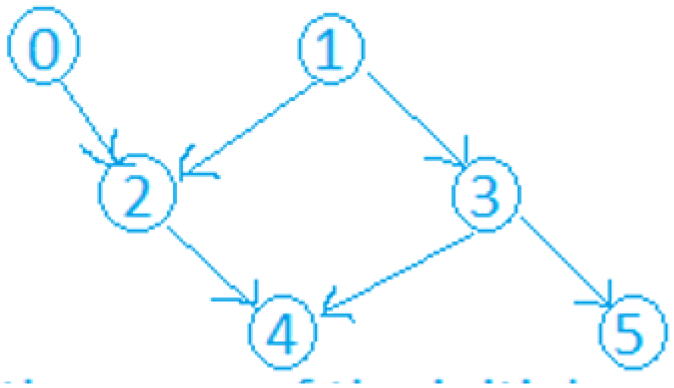
Part 2: Report

The DAG implementation is from https://github.com/thieman/py-dag/tree/master/dag.

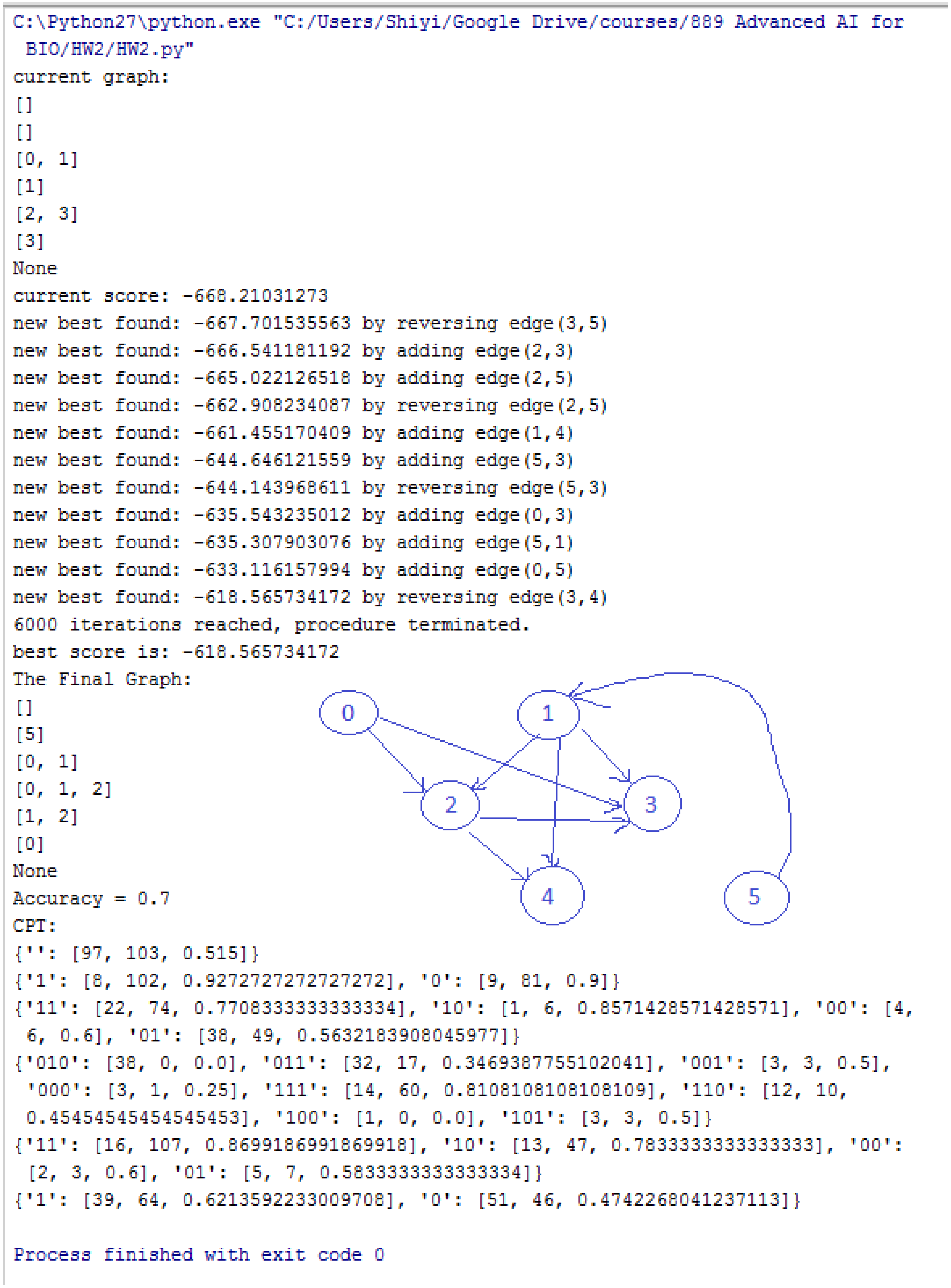
The reason I change the stopping criteria is that sometimes the score is not improved after an edge operation does not mean that the score will not improve in future operations. So I set up a max-iteration value as the stopping criteria. When set it to a number < 6000, the best score may vary each time the program runs. However, when set the max iteration to a number > 10000, most of the time, the score converges to -570.

The default initial graph is the same in all five trainings.

Initial graph:

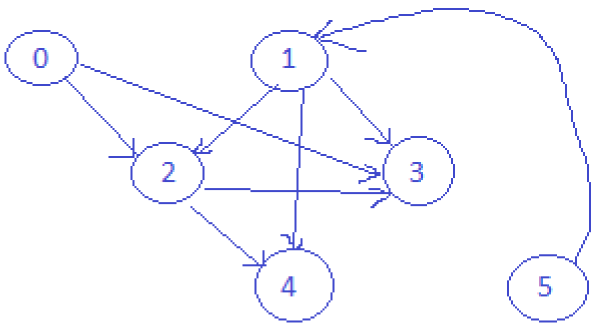


I have shown one graph output in README as an example. Here are the rest four graphs:



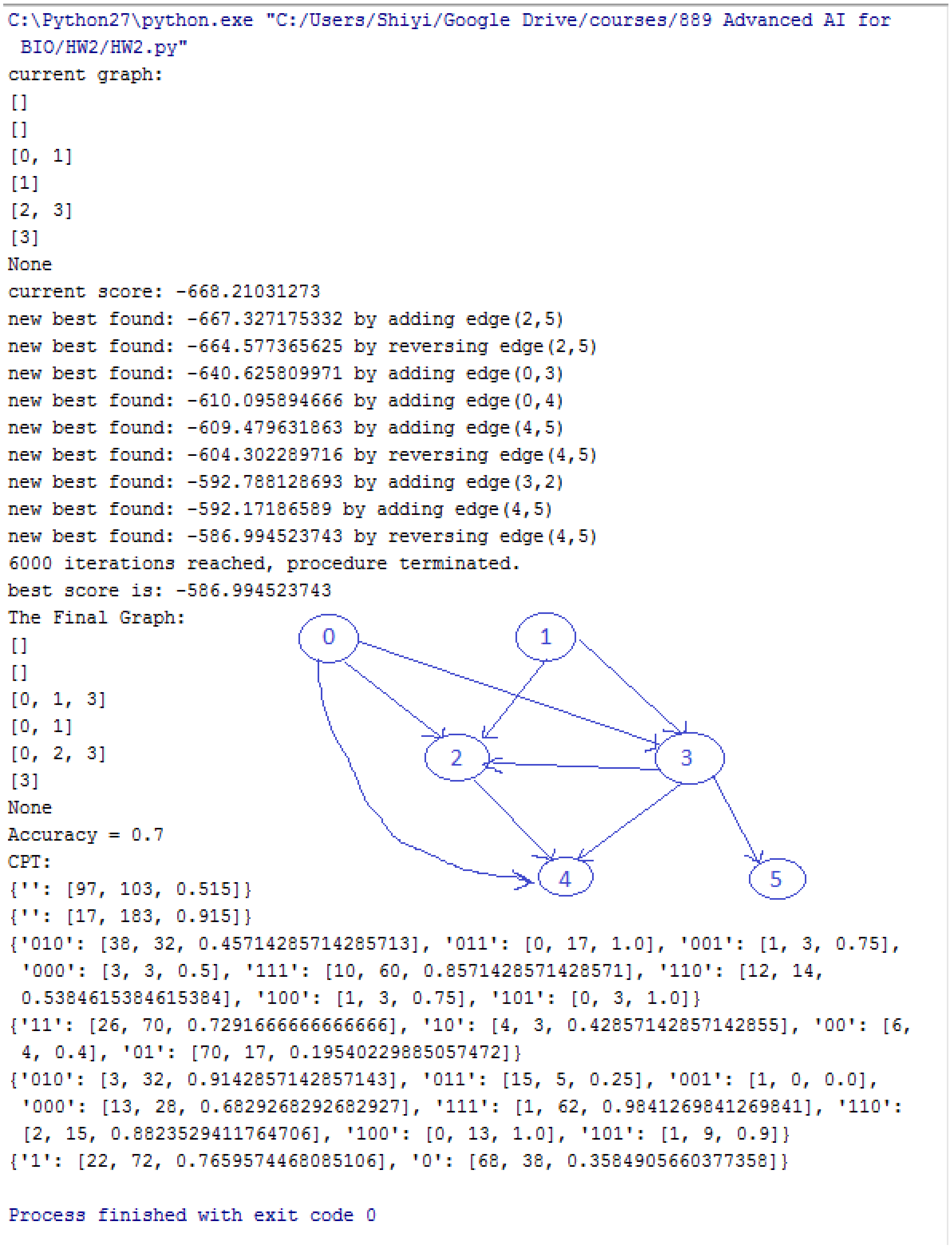
```
C:\Python27\python.exe "C:/Users/Shiyi/Google Drive/courses/889 Advanced AI for
 BIO/HW2/HW2.py"
current graph:
[]
[]
[0, 1]
[1]
[2, 3]
[3]
None
current score: -668.21031273
new best found: -667.701535563 by reversing edge(3,5)
new best found: -666.541181192 by adding edge(2,3)
new best found: -665.022126518 by adding edge(2,5)
new best found: -662.908234087 by reversing edge(2,5)
new best found: -661.455170409 by adding edge(1,4)
new best found: -644.646121559 by adding edge(5,3)
new best found: -644.143968611 by reversing edge(5,3)
new best found: -635.543235012 by adding edge(0,3)
new best found: -635.307903076 by adding edge(5,1)
new best found: -633.116157994 by adding edge(0,5)
new best found: -618.565734172 by reversing edge(3,4)
6000 iterations reached, procedure terminated.
best score is: -618.565734172
The Final Graph:
[]
[5]
[0, 1]
[0, 1, 2]
[1, 2]
[0]
None
Accuracy = 0.7
CPT:
{'': [97, 103, 0.515]}
{'1': [8, 102, 0.9272727272727272], '0': [9, 81, 0.9]}
{'11': [22, 74, 0.7708333333333334], '10': [1, 6, 0.8571428571428571], '00': [4,
 6, 0.6], '01': [38, 49, 0.5632183908045977]}
{'010': [38, 0, 0.0], '011': [32, 17, 0.3469387755102041], '001': [3, 3, 0.5],
 '000': [3, 1, 0.25], '111': [14, 60, 0.8108108108108109], '110': [12, 10,
 0.45454545454545453], '100': [1, 0, 0.0], '101': [3, 3, 0.5]}
{'11': [16, 107, 0.8699186991869918], '10': [13, 47, 0.7833333333333333], '00':
 [2, 3, 0.6], '01': [5, 7, 0.5833333333333334]}
{'1': [39, 64, 0.6213592233009708], '0': [51, 46, 0.4742268041237113]}

Process finished with exit code 0
```



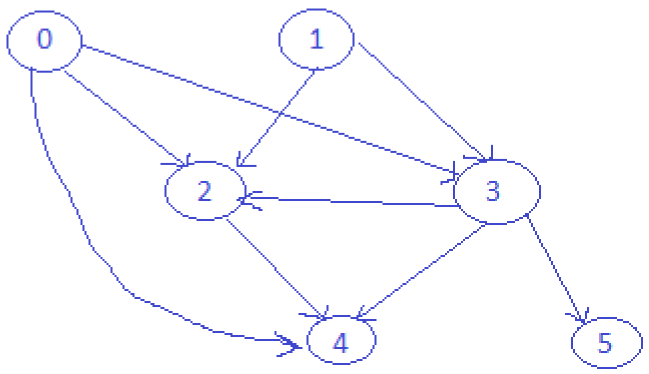
$$\begin{tikzcd}
0 \arrow[dr] \arrow[drrr] & & 1 \arrow[dl] \arrow[dr] \arrow[dd] & & \\
& 2 \arrow[rr] \arrow[dr] & & 3 & \\
& & 4 & & 5 \arrow[uull, bend right]
\end{tikzcd}$$

11



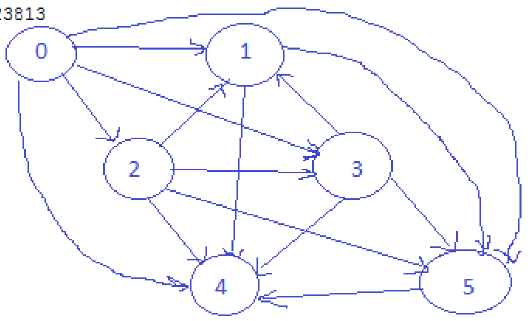
```
C:\Python27\python.exe "C:/Users/Shiyi/Google Drive/courses/889 Advanced AI for
 BIO/HW2/HW2.py"
current graph:
[]
[]
[0, 1]
[1]
[2, 3]
[3]
None
current score: -668.21031273
new best found: -667.327175332 by adding edge(2,5)
new best found: -664.577365625 by reversing edge(2,5)
new best found: -640.625809971 by adding edge(0,3)
new best found: -610.095894666 by adding edge(0,4)
new best found: -609.479631863 by adding edge(4,5)
new best found: -604.302289716 by reversing edge(4,5)
new best found: -592.788128693 by adding edge(3,2)
new best found: -592.17186589 by adding edge(4,5)
new best found: -586.994523743 by reversing edge(4,5)
6000 iterations reached, procedure terminated.
best score is: -586.994523743
The Final Graph:
[]
[]
[0, 1, 3]
[0, 1]
[0, 2, 3]
[3]
None
Accuracy = 0.7
CPT:
{'': [97, 103, 0.515]}
{'': [17, 183, 0.915]}
{'010': [38, 32, 0.45714285714285713], '011': [0, 17, 1.0], '001': [1, 3, 0.75],
 '000': [3, 3, 0.5], '111': [10, 60, 0.8571428571428571], '110': [12, 14,
 0.5384615384615384], '100': [1, 3, 0.75], '101': [0, 3, 1.0]}
{'11': [26, 70, 0.7291666666666666], '10': [4, 3, 0.42857142857142855], '00': [6,
 4, 0.4], '01': [70, 17, 0.19540229885057472]}
{'010': [3, 32, 0.9142857142857143], '011': [15, 5, 0.25], '001': [1, 0, 0.0],
 '000': [13, 28, 0.6829268292682927], '111': [1, 62, 0.9841269841269841], '110':
 [2, 15, 0.8823529411764706], '100': [0, 13, 1.0], '101': [1, 9, 0.9]}
{'1': [22, 72, 0.7659574468085106], '0': [68, 38, 0.3584905660377358]}

Process finished with exit code 0
```



```
C:\Python27\python.exe "C:/Users/Shiyi/Google Drive/courses/889 Advanced AI for
 BIO/HW2/HW2.py"
current graph:
[]
[]
[0, 1]
[1]
[2, 3]
[3]
None
current score: -668.21031273
new best found: -663.124301946 by adding edge(1,4)
new best found: -662.726817481 by adding edge(0,1)
new best found: -631.753915786 by adding edge(0,4)
new best found: -612.77358245 by adding edge(2,3)
new best found: -612.029473347 by adding edge(1,5)
new best found: -603.567481712 by adding edge(5,4)
new best found: -603.567481712 by reversing edge(1,2)
new best found: -601.86522435 by reversing edge(1,3)
new best found: -582.488194008 by adding edge(0,3)
new best found: -581.179818339 by adding edge(2,5)
new best found: -575.155171505 by adding edge(0,5)
new best found: -574.976211752 by adding edge(2,1)
new best found: -570.322523813 by adding edge(3,1)
6000 iterations reached, procedure terminated.
best score is: -570.322523813
The Final Graph:
[]
[0, 2, 3]
[0]
[0, 2]
[0, 1, 2, 3, 5]
[0, 1, 2, 3]
None
Accuracy = 0.75
CPT:
{'': [97, 103, 0.515]}
{'010': [3, 32, 0.9142857142857143], '011': [3, 17, 0.85], '001': [1, 0, 0.0],
 '000': [3, 38, 0.926829268292683], '111': [3, 60, 0.9523809523809523], '110':
 [3, 14, 0.8235294117647058], '100': [1, 12, 0.9230769230769231], '101': [0, 10,
 1.0]}
{'1': [23, 80, 0.7766990291262136], '0': [42, 55, 0.5670103092783505]}
{'11': [17, 63, 0.7875], '10': [13, 10, 0.43478260869565216], '00': [41, 1,
 0.023809523809523808], '01': [35, 20, 0.36363636363636365]}
{'10000': [0, 1, 1.0], '10100': [2, 1, 0.3333333333333333], '01101': [0, 14,
 1.0], '01100': [3, 15, 0.8333333333333334], '11111': [0, 45, 1.0], '11110': [1,
 14, 0.9333333333333333], '11010': [0, 1, 1.0], '11011': [1, 8, 0
 .8888888888888888], '01000': [8, 18, 0.6923076923076923], '01001': [4, 8,
 0.6666666666666666], '00111': [1, 0, 0.0], '00110': [2, 0, 0.0], '00011': [1, 0,
 0.0], '10110': [0, 1, 1.0], '10111': [0, 2, 1.0], '01110': [0, 3, 1.0], '01111':
 [12, 2, 0.14285714285714285], '11001': [0, 6, 1.0], '11000': [0, 6, 1.0],
 '11100': [0, 12, 1.0], '11101': [0, 2, 1.0], '00001': [1, 1, 0.5], '00000': [0,
 1, 1.0], '00100': [0, 1, 1.0], '00101': [0, 2, 1.0]}
{'0110': [18, 14, 0.4375], '0111': [3, 14, 0.8235294117647058], '0000': [1, 2,
 0.6666666666666666], '0001': [0, 1, 1.0], '0011': [2, 1, 0.3333333333333333],
 '0010': [1, 2, 0.6666666666666666], '0100': [26, 12, 0.3157894736842105],
 '1111': [15, 45, 0.75], '1110': [12, 2, 0.14285714285714285], '1100': [6, 6,
 0.5], '1101': [1, 9, 0.9], '1010': [3, 0, 0.0], '1011': [1, 2, 0
 .6666666666666666], '1000': [1, 0, 0.0]}

Process finished with exit code 0
```



```
C:\Python27\python.exe "C:/Users/Shiyi/Google Drive/courses/889 Advanced AI for
 BIO/HW2/HW2.py"
current graph:
[]
[]
[0, 1]
[1]
[2, 3]
[3]
None
current score: -668.21031273
new best found: -666.018567648 by adding edge(5,0)
new best found: -660.932556865 by adding edge(1,4)
new best found: -636.87774926 by adding edge(3,0)
new best found: -631.855732826 by adding edge(5,4)
new best found: -631.346955659 by reversing edge(3,5)
new best found: -614.754057729 by adding edge(0,4)
new best found: -597.44285593 by adding edge(3,5)
new best found: -596.934078764 by reversing edge(3,5)
6000 iterations reached, procedure terminated.
best score is: -596.934078764
The Final Graph:
[3, 5]
[]
[0, 1]
[1]
[0, 1, 2, 3, 5]
[]
None
Accuracy = 0.75
CPT:
{'11': [16, 56, 0.7777777777777778], '10': [5, 17, 0.7727272727272727], '00':
 [46, 22, 0.3235294117647059], '01': [30, 8, 0.21052631578947367]}
{'': [17, 183, 0.915]}
{'11': [22, 74, 0.7708333333333334], '10': [1, 6, 0.8571428571428571], '00': [4,
 6, 0.6], '01': [38, 49, 0.5632183908045977]}
{'1': [96, 87, 0.47540983606557374], '0': [10, 7, 0.4117647058823529]}
{'10000': [0, 1, 1.0], '10100': [2, 1, 0.3333333333333333], '01101': [0, 14,
 1.0], '01100': [3, 15, 0.8333333333333334], '11111': [0, 45, 1.0], '11110': [1,
 14, 0.9333333333333333], '11010': [0, 1, 1.0], '11011': [1, 8, 0
 .8888888888888888], '01000': [8, 18, 0.6923076923076923], '01001': [4, 8,
 0.6666666666666666], '00111': [1, 0, 0.0], '00110': [2, 0, 0.0], '00011': [1, 0,
 0.0], '10110': [0, 1, 1.0], '10111': [0, 2, 1.0], '01110': [0, 3, 1.0], '01111':
 [12, 2, 0.14285714285714285], '11001': [0, 6, 1.0], '11000': [0, 6, 1.0],
 '11100': [0, 12, 1.0], '11101': [0, 2, 1.0], '00001': [1, 1, 0.5], '00000': [0,
 1, 1.0], '00100': [0, 1, 1.0], '00101': [0, 2, 1.0]}
{'': [90, 110, 0.55]}

Process finished with exit code 0
```