Carlos Mariscal
ECE 528
Class project

The purpose of this project is to implement the Kl partitioning in software. I decided to use python. The software takes in an input, which describes a graph with nodes and edges. Below are instructions on how to run the software.

The algorithm was applied to 2 small examples (shown below)by hand and then the results are shown with the software. Both hand and software results match.

The output of the class example graph is also shown below. The final cost of partition is 9.

Instructions:

source.py tested in python 2.7.10 and it should work with most versions of python but is not guaranteed. It will not work with python 2.4.3, which is installed in the MCECS Linux servers. In this case run source_for_mcecs.py.
to find out which version of python is installed in the system type "python -V"

In order to run from command line in Linux or in Mac terminal type. (assuming source.py and input are in the present working directory)

```
python source.py  inputfile [-f]
```

Or for mcecs

```
python source_for_mces.py inputfile [-f]
```

The ,[-f] option is optional it should only be used for large files with large node count.

Example:

```
python source.py KLinput20F2015.txt
```

this will run the class example

```
python source.py input_2851nodes.txt -f
```

this will run the file with 2851 node (of course you need to name it this in your system) with the fast option

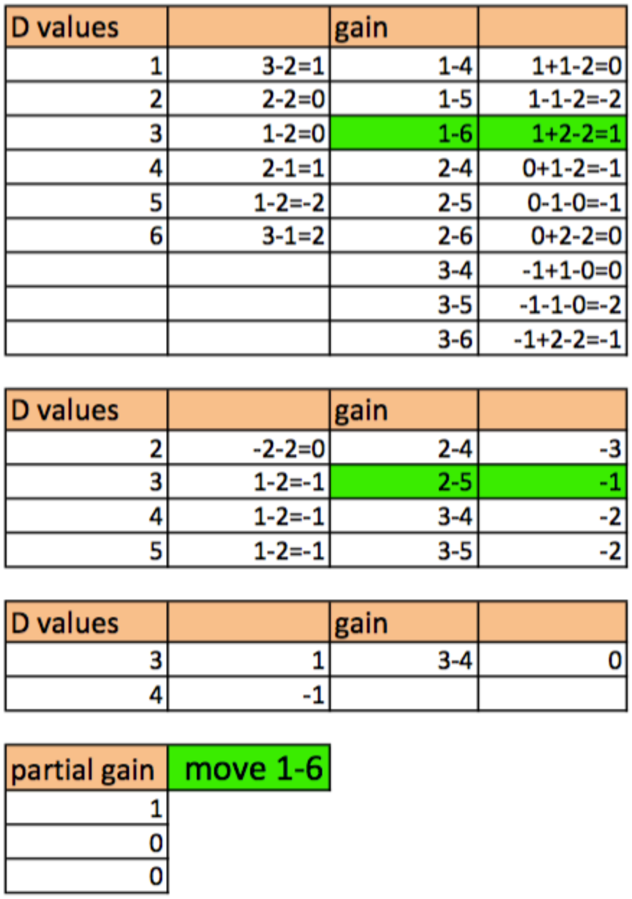
Carlos Mariscal
ECE 528
KL partitioning report

Example 1

```
6 11
2 3 4 5 6
1 3 4 6
1 2 6
1 2 5
1 4 6
1 2 3 5
```

output

```
Iteration  1
Partition 1 [1, 2, 3]
Partition 2 [4, 5, 6]
Cost of Partition 6
Iteration  2
Partition 1 [2, 3, 6]
Partition 2 [5, 4, 1]
Cost of Partition 5
********
all done
********
Partition 1
[2, 3, 6]
Partition 2
[1, 4, 5]
Cost of partition 5
```



D values		gain	
1	3-2=1	1-4	1+1-2=0
2	2-2=0	1-5	1-1-2=-2
3	1-2=0	1-6	1+2-2=1
4	2-1=1	2-4	0+1-2=-1
5	1-2=-2	2-5	0-1-0=-1
6	3-1=2	2-6	0+2-2=0
		3-4	-1+1-0=0
		3-5	-1-1-0=-2
		3-6	-1+2-2=-1

D values		gain	
2	-2-2=0	2-4	-3
3	1-2=-1	2-5	-1
4	1-2=-1	3-4	-2
5	1-2=-1	3-5	-2

D values		gain	
3	1	3-4	0
4	-1		

partial gain	move 1-6
1	
0	
0	

Figure 1 hand calculated results for example 1

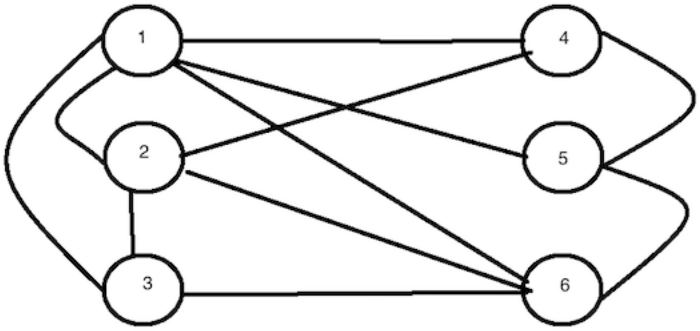


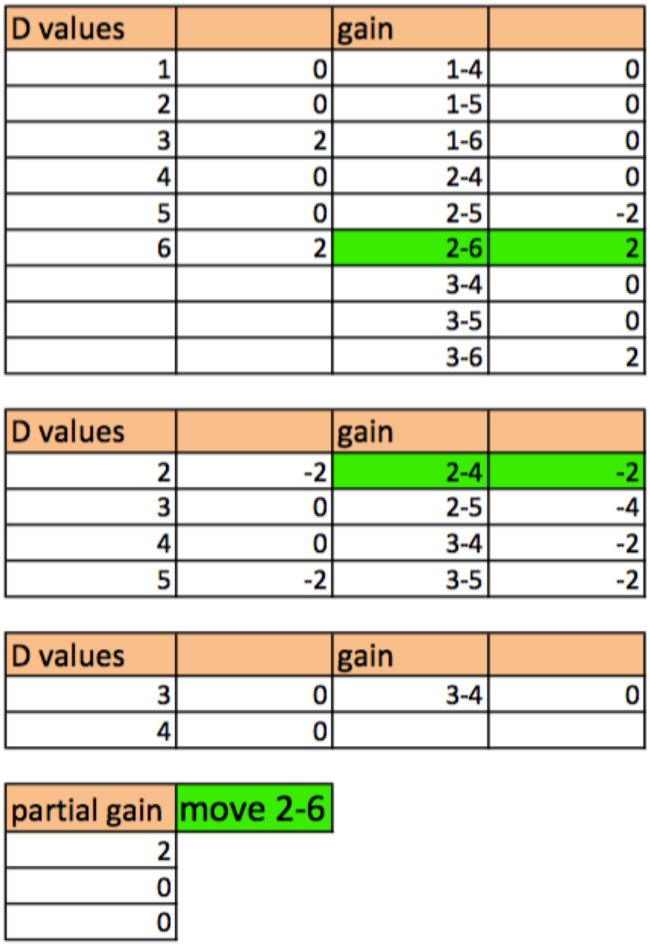
Figure 2 graph of example 1

Example 2

```
6 7
3 6
3 5
1 2 4 6
3 5
2 4
1 3
```

Output

```
Iteration  1
Partition 1 [1, 2, 3]
Partition 2 [4, 5, 6]
Cost of Partition 4
Iteration  2
Partition 1 [1, 3, 6]
Partition 2 [4, 5, 2]
Cost of Partition 2
********
all done
********
Partition 1
[1, 3, 6]
Partition 2
[2, 4, 5]
Cost of partition 2
```



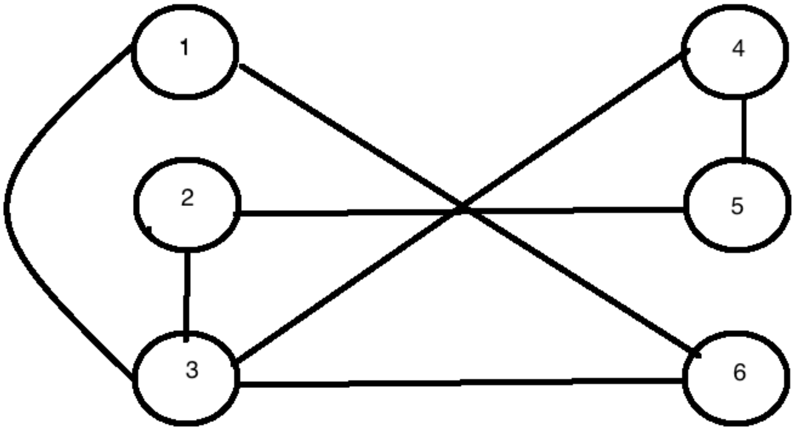
D values		gain	
1	0	1-4	0
2	0	1-5	0
3	2	1-6	0
4	0	2-4	0
5	0	2-5	-2
6	2	2-6	2
		3-4	0
		3-5	0
		3-6	2

D values		gain	
2	-2	2-4	-2
3	0	2-5	-4
4	0	3-4	-2
5	-2	3-5	-2

D values		gain	
3	0	3-4	0
4	0		

partial gain	move 2-6
2	
0	
0	

Figure 3 hand calculated results for example 2



KL class example output

```
Iteration  1
Partition 1 [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
Partition 2 [21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40]
Cost of Partition 38

Iteration  2
Partition 1 [7, 3, 14, 5, 2, 8, 17, 10, 20, 6, 13, 4, 15, 40, 24, 37, 35, 38, 21, 25]
Partition 2 [34, 32, 36, 30, 28, 23, 26, 22, 31, 27, 29, 39, 33, 12, 18, 1, 9, 11, 19, 16]
Cost of Partition 22

Iteration  3
Partition 1 [8, 10, 15, 4, 38, 20, 6, 40, 13, 2, 34, 32, 36, 30, 23, 26, 28, 18, 12, 22]
Partition 2 [1, 27, 31, 29, 33, 9, 11, 39, 19, 16, 37, 14, 7, 3, 21, 25, 17, 35, 24, 5]
Cost of Partition 9

********
all done
********
Partition 1
[2, 4, 6, 8, 10, 12, 13, 15, 18, 20, 22, 23, 26, 28, 30, 32, 34, 36, 38, 40]
Partition 2
[1, 3, 5, 7, 9, 11, 14, 16, 17, 19, 21, 24, 25, 27, 29, 31, 33, 35, 37, 39]
Cost of partition 9
```