Purpose: To implement Kruskal’s and Dijkstra’s Algorithm and to implement Heap Sort
Method:
Language: Python 3
Compilation Instructions:

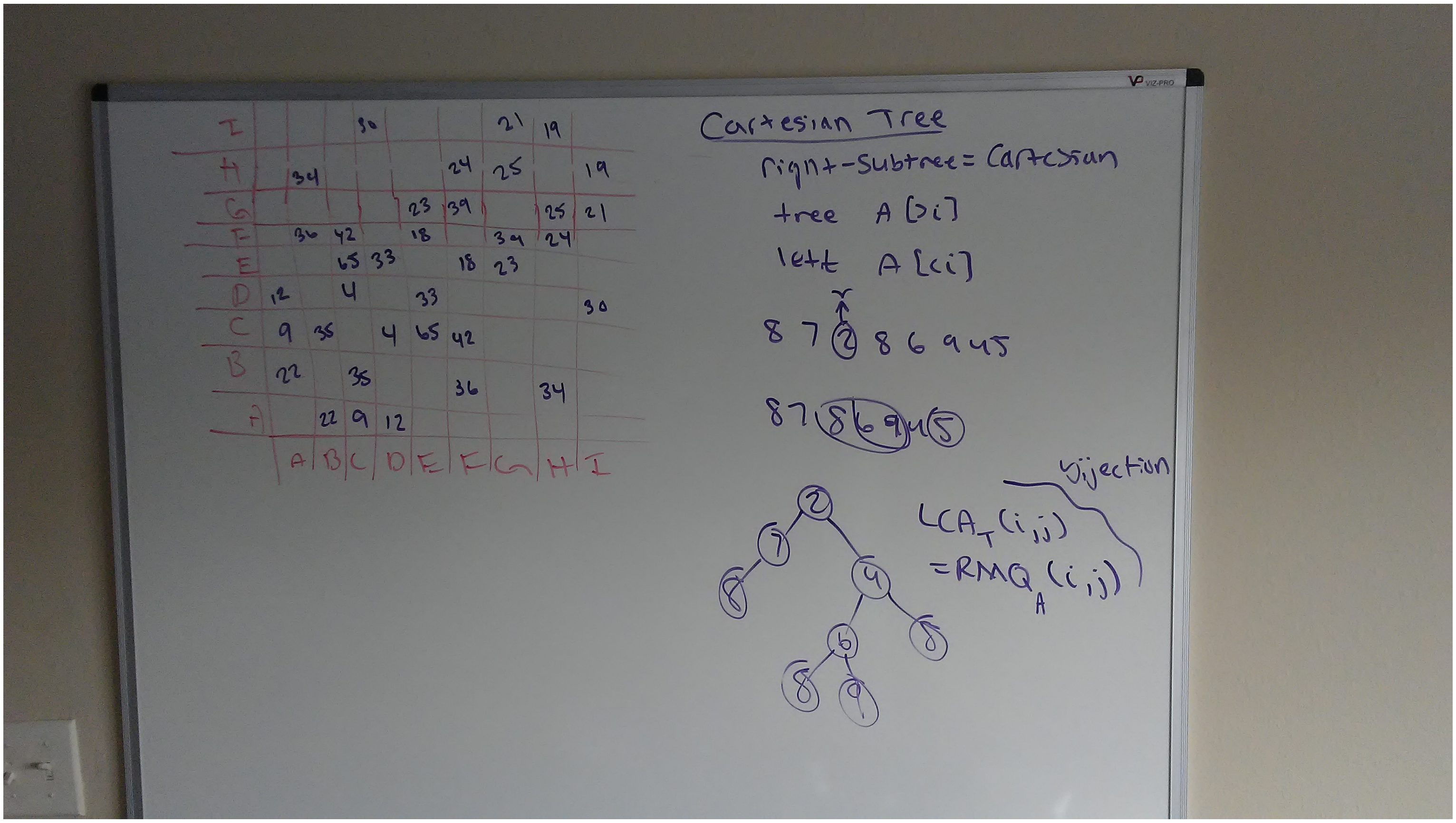
Very simple. To run on Eustis, call “python __init__.py”
Outputs are below
Discussion

Heapsort: O(nlogn + nlogn) The extra nlogn is there because I am inserting the elements into the heap one at a time and then allowing the heap to maintain the heap order automatically after inserting the element. Thus, the heap calls heapifyup after each insert which is O(logn). Iterating over the list is O(n). Thus, the extra factor of O(nlogn).

Kruskal’s: O(mlogn), m = number of edges, n = number of vertices

Dijkstra’s: O(mlogn), m = number of edges, n = number of vertices

[A] : (B|22), (C|9), (D|12)
[B] : (A|22), (C|35), (F|36), (H|34)
[C] : (A|9), (B|35), (D|4), (E|65), (F|42)
[D] : (A|12), (C|4), (E|33), (I|30)
[E] : (C|65), (D|33), (F|18), (G|23)
[F] : (B|36), (C|42), (E|18), (G|39), (H|24)
[G] : (E|23), (F|39), (H|25), (I|21)
[H] : (B|34), (F|24), (G|25), (I|19)
[I] : (D|30), (G|21), (H|19)



I				30			21	19	
H		34				24	25		19
G					23	39		25	21
F		36	42		18		39	24	
E			65	33		18	23		
D	12		4		33				30
C	9	35		4	65	42			
B	22		35			36		34	
A		22	9	12					
	A	B	C	D	E	F	G	H	I

Cartesian Tree

right-subtree = Cartesian tree A[>i]

left A[<i]

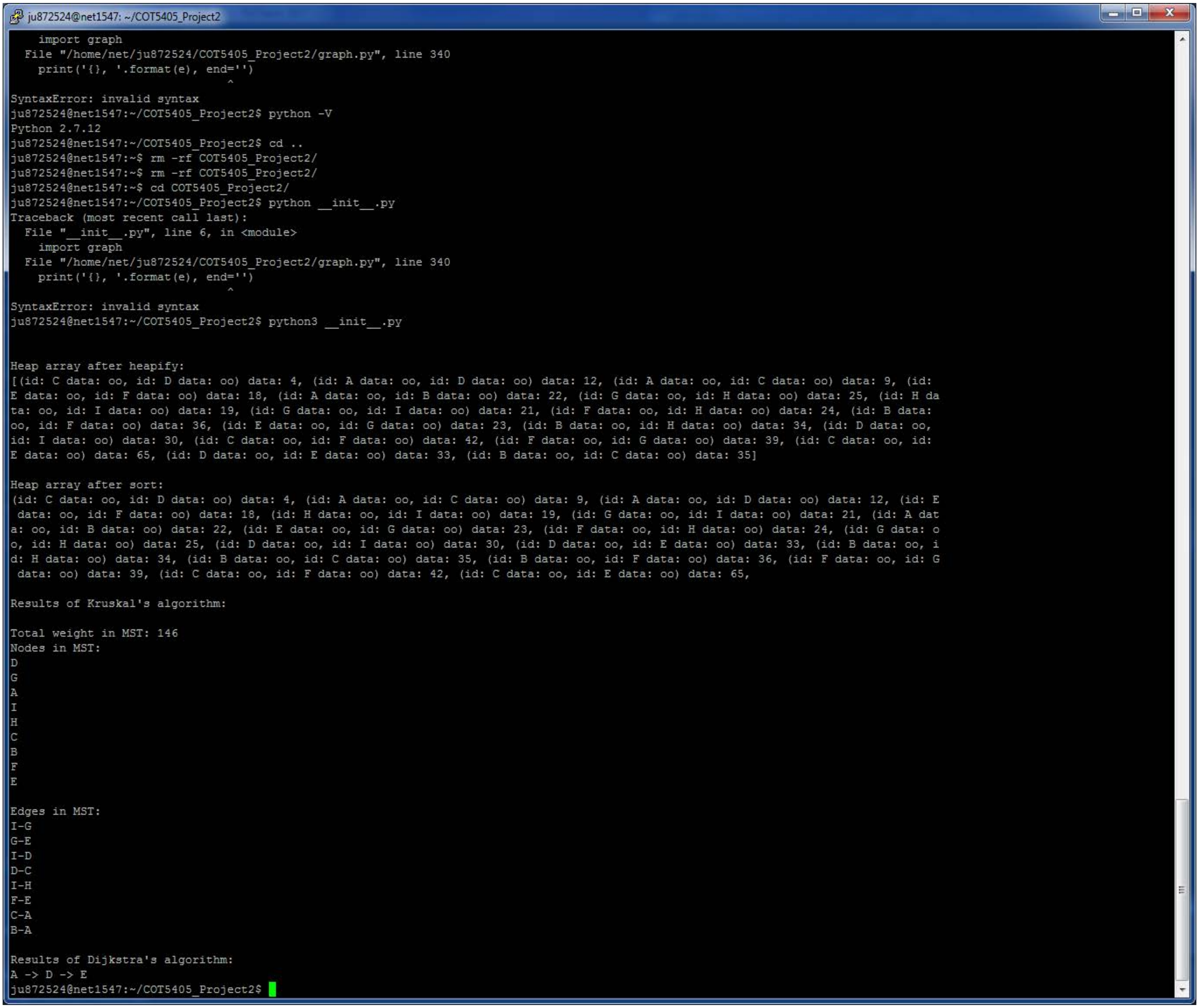
r

8 7 2 8 6 9 4 5

8 7 8 6 9 4 5

bijection

$LCA_T(i,j) = RMQ_A(i,j)$



```
    import graph
  File "/home/net/ju872524/COT5405_Project2/graph.py", line 340
    print('{}, '.format(e), end='')
                                ^
SyntaxError: invalid syntax
ju872524@net1547:~/COT5405_Project2$ python -V
Python 2.7.12
ju872524@net1547:~/COT5405_Project2$ cd ..
ju872524@net1547:~$ rm -rf COT5405_Project2/
ju872524@net1547:~$ rm -rf COT5405_Project2/
ju872524@net1547:~$ cd COT5405_Project2/
ju872524@net1547:~/COT5405_Project2$ python __init__.py
Traceback (most recent call last):
  File "__init__.py", line 6, in <module>
    import graph
  File "/home/net/ju872524/COT5405_Project2/graph.py", line 340
    print('{}, '.format(e), end='')
                                ^
SyntaxError: invalid syntax
ju872524@net1547:~/COT5405_Project2$ python3 __init__.py


Heap array after heapify:
[(id: C data: oo, id: D data: oo) data: 4, (id: A data: oo, id: D data: oo) data: 12, (id: A data: oo, id: C data: oo) data: 9, (id:
E data: oo, id: F data: oo) data: 18, (id: A data: oo, id: B data: oo) data: 22, (id: G data: oo, id: H data: oo) data: 25, (id: H da
ta: oo, id: I data: oo) data: 19, (id: G data: oo, id: I data: oo) data: 21, (id: F data: oo, id: H data: oo) data: 24, (id: B data:
oo, id: F data: oo) data: 36, (id: E data: oo, id: G data: oo) data: 23, (id: B data: oo, id: H data: oo) data: 34, (id: D data: oo,
id: I data: oo) data: 30, (id: C data: oo, id: F data: oo) data: 42, (id: F data: oo, id: G data: oo) data: 39, (id: C data: oo, id:
E data: oo) data: 65, (id: D data: oo, id: E data: oo) data: 33, (id: B data: oo, id: C data: oo) data: 35]

Heap array after sort:
(id: C data: oo, id: D data: oo) data: 4, (id: A data: oo, id: C data: oo) data: 9, (id: A data: oo, id: D data: oo) data: 12, (id: E
 data: oo, id: F data: oo) data: 18, (id: H data: oo, id: I data: oo) data: 19, (id: G data: oo, id: I data: oo) data: 21, (id: A dat
a: oo, id: B data: oo) data: 22, (id: E data: oo, id: G data: oo) data: 23, (id: F data: oo, id: H data: oo) data: 24, (id: G data: o
o, id: H data: oo) data: 25, (id: D data: oo, id: I data: oo) data: 30, (id: D data: oo, id: E data: oo) data: 33, (id: B data: oo, i
d: H data: oo) data: 34, (id: B data: oo, id: C data: oo) data: 35, (id: B data: oo, id: F data: oo) data: 36, (id: F data: oo, id: G
 data: oo) data: 39, (id: C data: oo, id: F data: oo) data: 42, (id: C data: oo, id: E data: oo) data: 65,

Results of Kruskal's algorithm:

Total weight in MST: 146
Nodes in MST:
D
G
A
I
H
C
B
F
E

Edges in MST:
I-G
G-E
I-D
D-C
I-H
F-E
C-A
B-A

Results of Dijkstra's algorithm:
A -> D -> E
ju872524@net1547:~/COT5405_Project2$
```

Text is below if you cannot read the screenshot

```
Heap array after heapify:
[(id: C data: oo, id: D data: oo) data: 4, (id: E data: oo, id: F data: oo) data: 18, (id: A data: oo, id: C data: oo) data:
9, (id: G data: oo, id: I data: oo) data: 21, (id: E data: oo, id: G data: oo) data: 23, (id: H data: oo, id: I data: oo) data:
19, (id: A data: oo, id: D data: oo) data: 12, (id: B data: oo, id: F data: oo) data: 36, (id: D data: oo, id: I data: oo)
data: 30, (id: B data: oo, id: C data: oo) data: 35, (id: G data: oo, id: H data: oo) data: 25, (id: A data: oo, id: B data:
oo) data: 22, (id: F data: oo, id: H data: oo) data: 24, (id: C data: oo, id: E data: oo) data: 65, (id: D data: oo, id: E
data: oo) data: 33, (id: C data: oo, id: F data: oo) data: 42, (id: F data: oo, id: G data: oo) data: 39, (id: B data: oo, id:
H data: oo) data: 34]

Heap array after sort:
(id: C data: oo, id: D data: oo) data: 4, (id: A data: oo, id: C data: oo) data: 9, (id: A data: oo, id: D data: oo) data:
12, (id: E data: oo, id: F data: oo) data: 18, (id: H data: oo, id: I data: oo) data: 19, (id: G data: oo, id: I data: oo)
data: 21, (id: A data: oo, id: B data: oo) data: 22, (id: E data: oo, id: G data: oo) data: 23, (id: F data: oo, id: H data:
oo) data: 24, (id: G data: oo, id: H data: oo) data: 25, (id: D data: oo, id: I data: oo) data: 30, (id: D data: oo, id: E
data: oo) data: 33, (id: B data: oo, id: H data: oo) data: 34, (id: B data: oo, id: C data: oo) data: 35, (id: B data: oo, id:
F data: oo) data: 36, (id: F data: oo, id: G data: oo) data: 39, (id: C data: oo, id: F data: oo) data: 42, (id: C data: oo,
id: E data: oo) data: 65,

Results of Kruskal's algorithm:

Total weight in MST: 146
Nodes in MST:
B
H
F
E
G
C
D
I
A

Edges in MST:
F-E
C-A
I-G
G-E
B-A
D-C
I-H
I-D

Results of Dijkstra's algorithm:
A -> D -> E
```