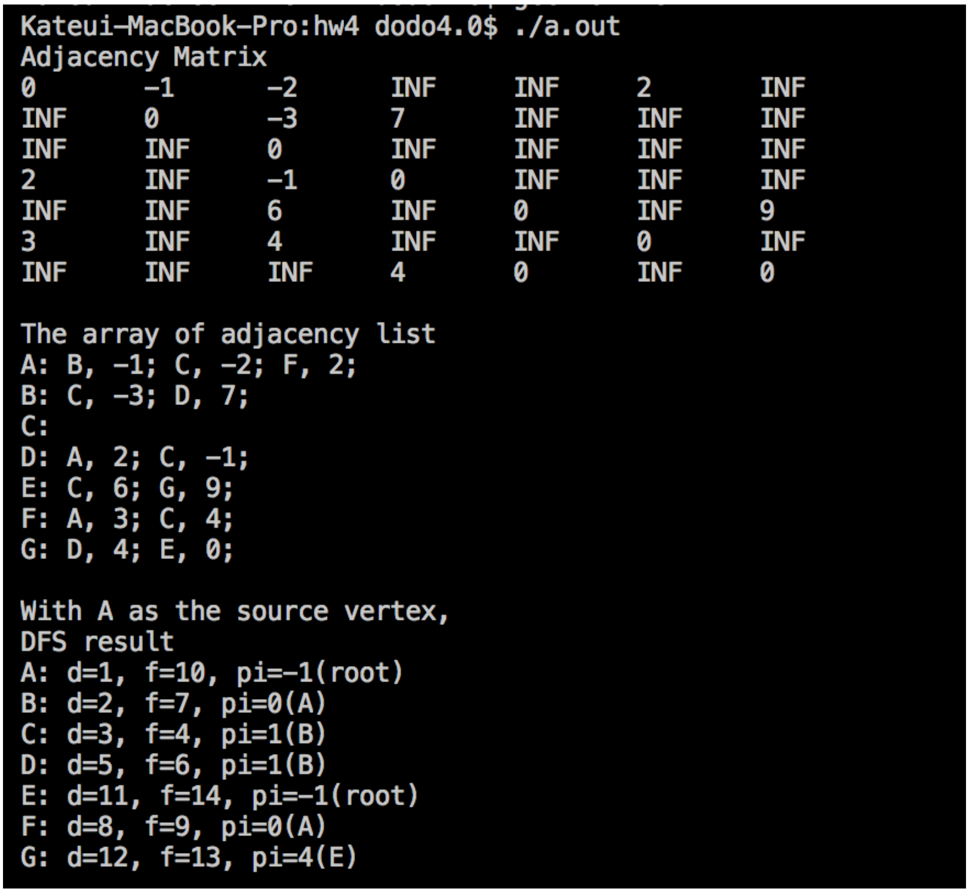
ITP30005 Algorithm Analysis Homework 4 Report

21700083 Kim Doyoon

1.

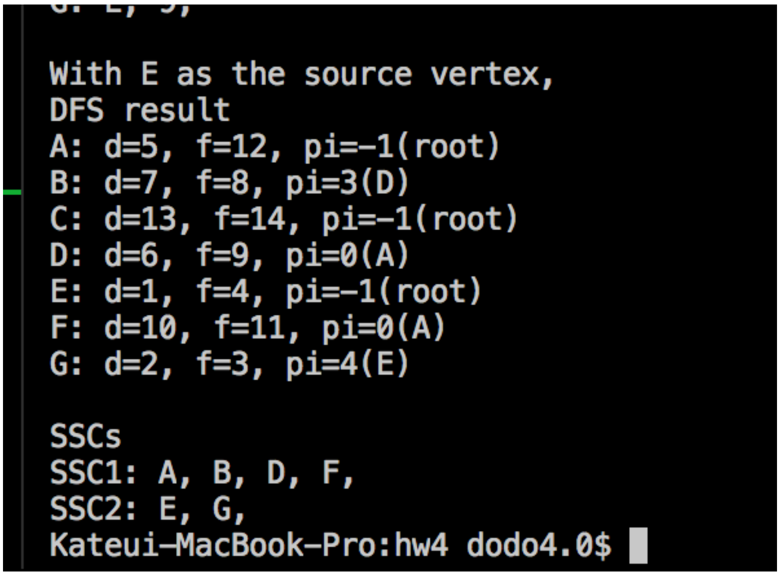


```
Kateui-MacBook-Pro:hw4 dodo4.0$ ./a.out
Adjacency Matrix
0       -1      -2      INF     INF     2       INF
INF     0       -3      7       INF     INF     INF
INF     INF     0       INF     INF     INF     INF
2       INF     -1      0       INF     INF     INF
INF     INF     6       INF     0       INF     9
3       INF     4       INF     INF     0       INF
INF     INF     INF     4       0       INF     0

The array of adjacency list
A: B, -1; C, -2; F, 2;
B: C, -3; D, 7;
C:
D: A, 2; C, -1;
E: C, 6; G, 9;
F: A, 3; C, 4;
G: D, 4; E, 0;

With A as the source vertex,
DFS result
A: d=1, f=10, pi=-1(root)
B: d=2, f=7, pi=0(A)
C: d=3, f=4, pi=1(B)
D: d=5, f=6, pi=1(B)
E: d=11, f=14, pi=-1(root)
F: d=8, f=9, pi=0(A)
G: d=12, f=13, pi=4(E)
```

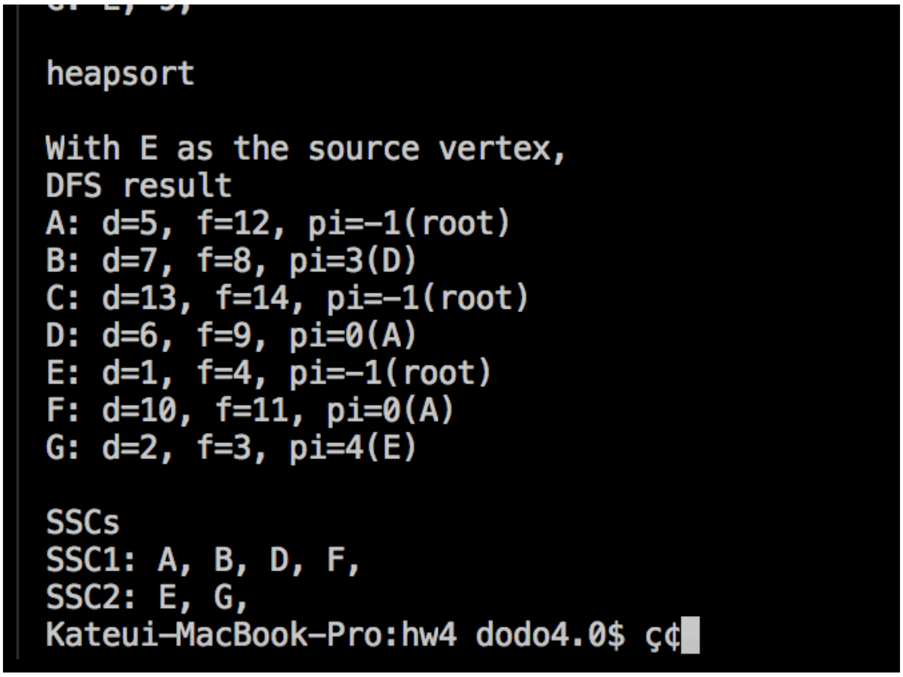
2.



```
With E as the source vertex,
DFS result
A: d=5, f=12, pi=-1(root)
B: d=7, f=8, pi=3(D)
C: d=13, f=14, pi=-1(root)
D: d=6, f=9, pi=0(A)
E: d=1, f=4, pi=-1(root)
F: d=10, f=11, pi=0(A)
G: d=2, f=3, pi=4(E)

SSCs
SSC1: A, B, D, F,
SSC2: E, G,
Kateui-MacBook-Pro:hw4 dodo4.0$
```

3.

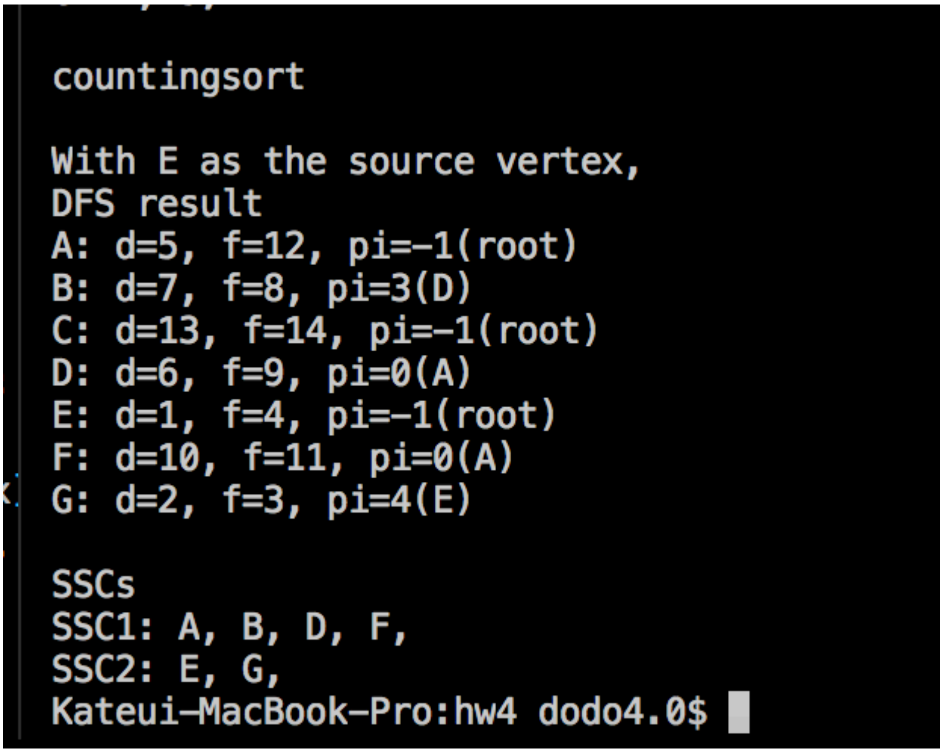


```
heapsort

With E as the source vertex,
DFS result
A: d=5, f=12, pi=-1(root)
B: d=7, f=8, pi=3(D)
C: d=13, f=14, pi=-1(root)
D: d=6, f=9, pi=0(A)
E: d=1, f=4, pi=-1(root)
F: d=10, f=11, pi=0(A)
G: d=2, f=3, pi=4(E)

SSCs
SSC1: A, B, D, F,
SSC2: E, G,
Kateui-MacBook-Pro:hw4 dodo4.0$ ç¢
```

4.



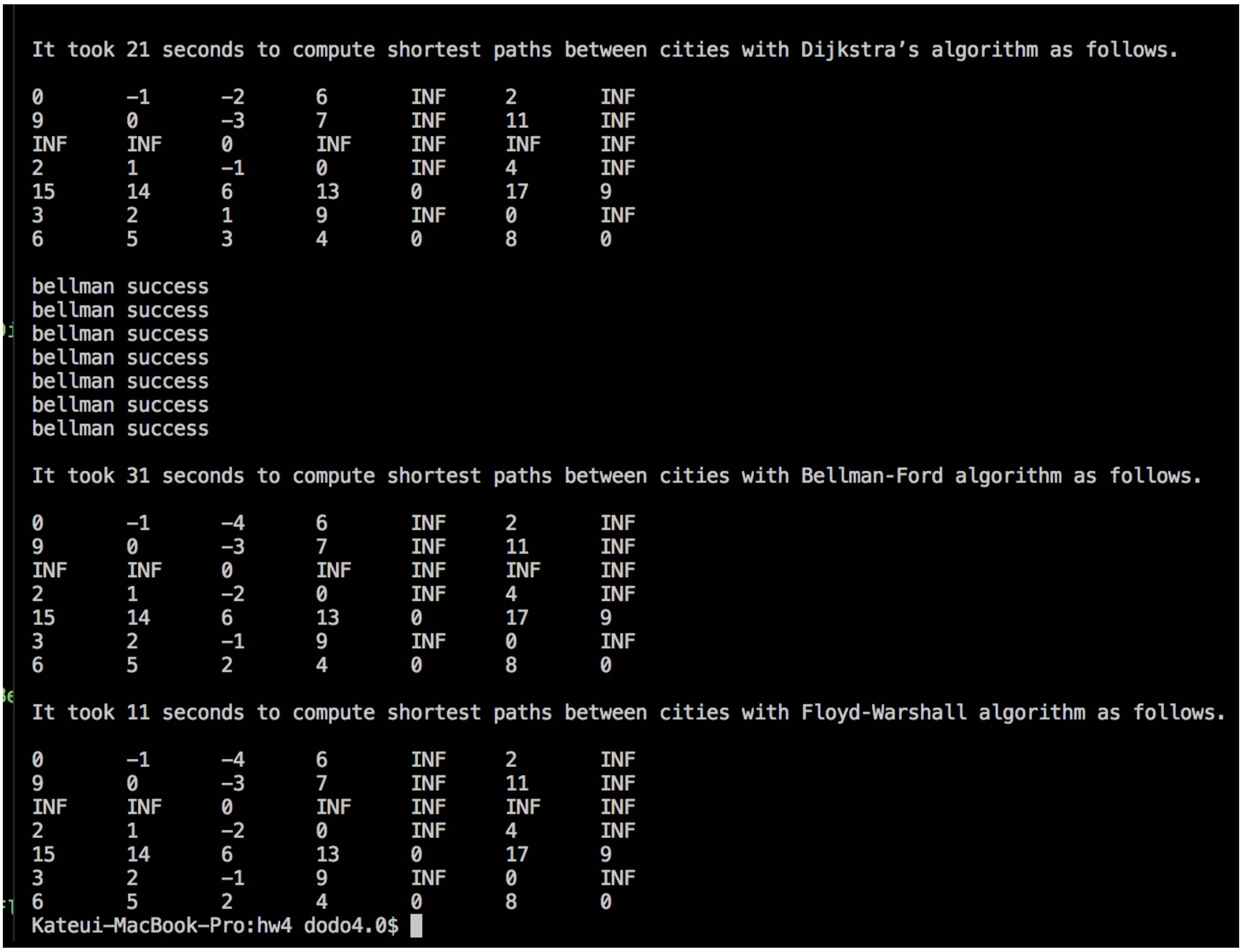
```
countingsort

With E as the source vertex,
DFS result
A: d=5, f=12, pi=-1(root)
B: d=7, f=8, pi=3(D)
C: d=13, f=14, pi=-1(root)
D: d=6, f=9, pi=0(A)
E: d=1, f=4, pi=-1(root)
F: d=10, f=11, pi=0(A)
G: d=2, f=3, pi=4(E)

SSCs
SSC1: A, B, D, F,
SSC2: E, G,
Kateui-MacBook-Pro:hw4 dodo4.0$
```

5.

(i) (ii) (iii)



```
It took 21 seconds to compute shortest paths between cities with Dijkstra's algorithm as follows.

0       -1      -2      6       INF     2       INF
9       0       -3      7       INF     11      INF
INF     INF     0       INF     INF     INF     INF
2       1       -1      0       INF     4       INF
15      14      6       13      0       17      9
3       2       1       9       INF     0       INF
6       5       3       4       0       8       0

bellman success
bellman success
bellman success
bellman success
bellman success
bellman success
bellman success

It took 31 seconds to compute shortest paths between cities with Bellman-Ford algorithm as follows.

0       -1      -4      6       INF     2       INF
9       0       -3      7       INF     11      INF
INF     INF     0       INF     INF     INF     INF
2       1       -2      0       INF     4       INF
15      14      6       13      0       17      9
3       2       -1      9       INF     0       INF
6       5       2       4       0       8       0

It took 11 seconds to compute shortest paths between cities with Floyd-Warshall algorithm as follows.

0       -1      -4      6       INF     2       INF
9       0       -3      7       INF     11      INF
INF     INF     0       INF     INF     INF     INF
2       1       -2      0       INF     4       INF
15      14      6       13      0       17      9
3       2       -1      9       INF     0       INF
6       5       2       4       0       8       0
Kateui-MacBook-Pro:hw4 dodo4.0$ 
```

(iv) Result using Bellman-Ford algorithm and Floyd-Warshall algorithm is the same, but the result using Dijkstra's algorithm is different. This is because Dijkstra's algorithm doesn't allow negative edge weights. This is because Dijkstra's algorithm is based on greedy method.