Tutorial 10.

1. Compute the strongly connected components of graphs (i) and (ii) below.

   (i) [B] [E] [A] [G H I] [C D F J]
   (ii) [A B E] [C] [D F G H I]

2. Use the graph reversal algorithm to detect if (i) and (ii) are strongly connected.

   For (i) the normal graph started on any node, e.g., A tells us that not all vertices are reachable. However, for (ii), starting from A, DFS visits everything. On reversal, A gives E and B. That's it. So this graph is not strongly connected.

3. Recall the definition of strong connectedness. Let us define v~w iff there is a path from v to w and a path from w to v. This is an equivalence relation on vertices. Let [v1]…[vk] be the equivalence classes of ~. We now define a new graph G'(V',E'), where V'={ [v1],...,[vk]}. We say ([vi],[vj]) is an edge in the new graph iff there is a path from vi to vj in the original graph.
   (A) Show that this new graph G' has no cycles.

   Note that for any two vertices w and w' within a given component, there is a path from w to w'. Note that if there is a path from vi to vj and wi in [vi] and wj in [vj] then there is a path from wi to wj in the original graph. This proves (B). Let us now prove (A)

   Suppose [v1]->[v2]->...->[vk]->[v1] is a cycle. Then there is a cycle w1->w2->...->wk->w1 in the original graph, with wi in [vi]. Then [v1]=v[2]=...[vk].

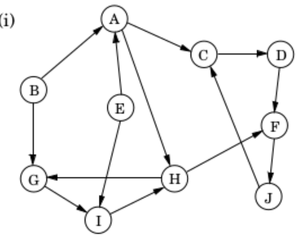
   (B) Show that the edge ([vi],[vj]) does not depend on the representative vi of [vi].
   (C) Compute G' for the first example graph:
   (i) [B]->[A], [E]->A, [A]->[G H I], [A]->{C D F J], [G H I]->[C D F J]

4. Run the smallest arrival time algorithm for the example graphs for DFS starting at A and proceeding lexicographically. If you have not exhausted all vertices, start at the next unvisited vertex in lexicographic order. Record this time in a table.

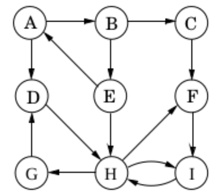
   Use the code supplied.

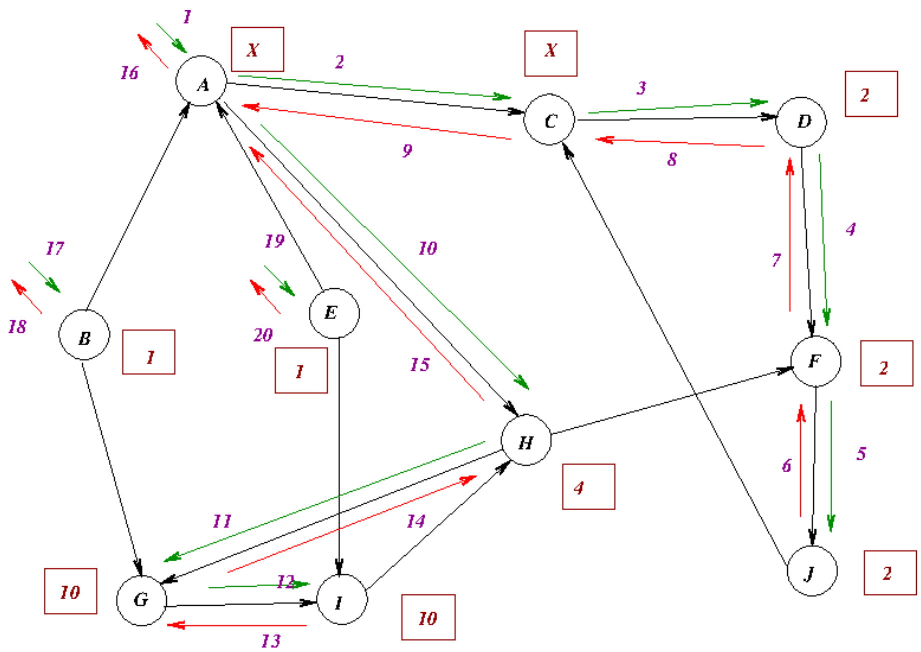
Vertex	In	Out	Min arrival time for edge going out for subtree rooted there
A	1	16	x
B	17	18	1
C	2	9	x
D	3	8	2
E	19	20	1
F	4	7	2
G	11	14	10
H	10	15	4
I	12	13	10
J	5	6	2

(i)


A
B
C
D
E
F
G
H
I
J

(ii)


A
B
C
D
E
F
G
H
I


A
B
C
D
E
F
G
H
I
J
X
X
2
2
2
1
1
4
10
10
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20