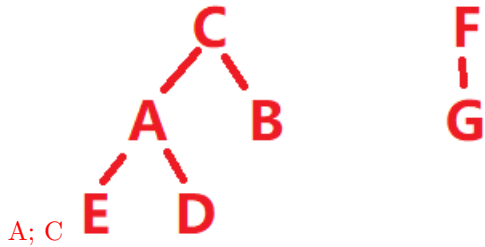


Problem 1(2+3pts): Consider a disjoint set with both path compression and union-by-size optimization. When two trees have the same height, the set specified first in the union will be the root of the merged set. The following operations are done:

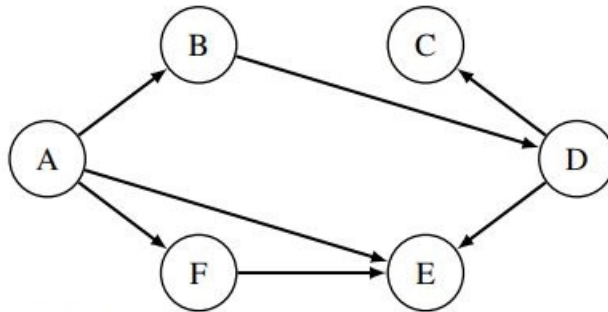
$union(A, D), find(A), union(C, B), union(E, D), union(F, G), union(C, A), find(B)$

Please write down the result for each *find* operation appeared in the above operations, and draw the disjoint set tree after all operations above is finished.



Problem 2(3pts):

Run BFS the following graph starting with vertex A. Please write down the result. Whenever there is a choice of which node to visit next, follow the alphabetical order.



A B E F D C

Problem 3(7pts): Maze

There is a maze stored in a $m \times n$ matrix A. If there is an obstacle at point (i, j) , $A[i][j]=1$; otherwise $A[i][j]=0$. You're at point $(1, 1)$, and you want to go to (m, n) .

	1	2	3	4	5
1	You				
2	1				
3	2				
4	3		7	8	9
5	4	5	6		Exit

Please design an algorithm to find out the shortest path from $(1, 1)$ to (m, n) . Briefly explain your algorithm using natural language. For the provided example, the result is 10.

The input matrix can be seen as a undirected graph, each point (i, j) as a vertex, and there are edges between each point with value 0 and each of its neighbor with value 0. Use BFS to search this graph starting at point $(1, 1)$. For each node appeared in the fringe, record its parent. When we reach point (m,n) , stop, then we can calculate the shortest path by the recorded parent.