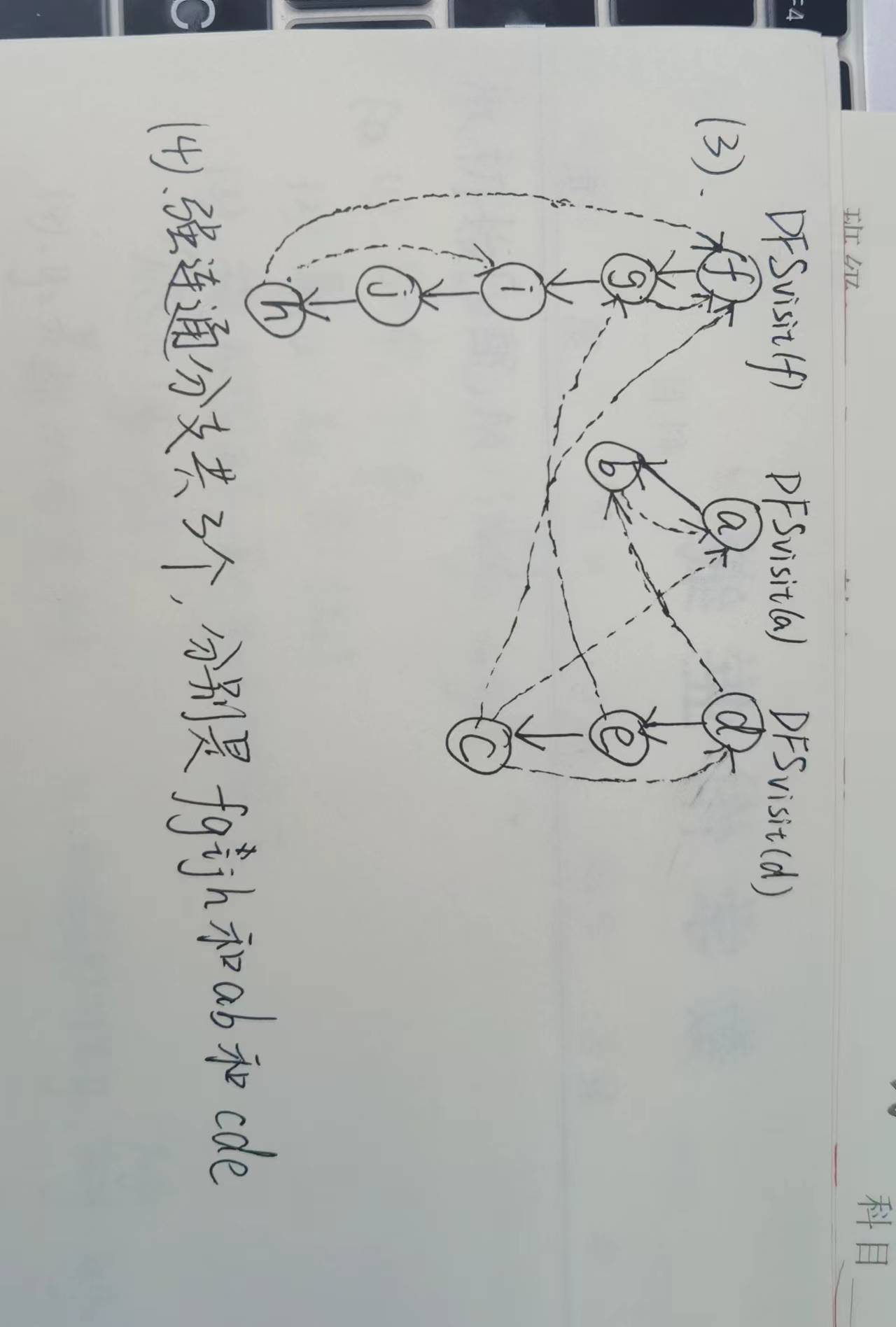
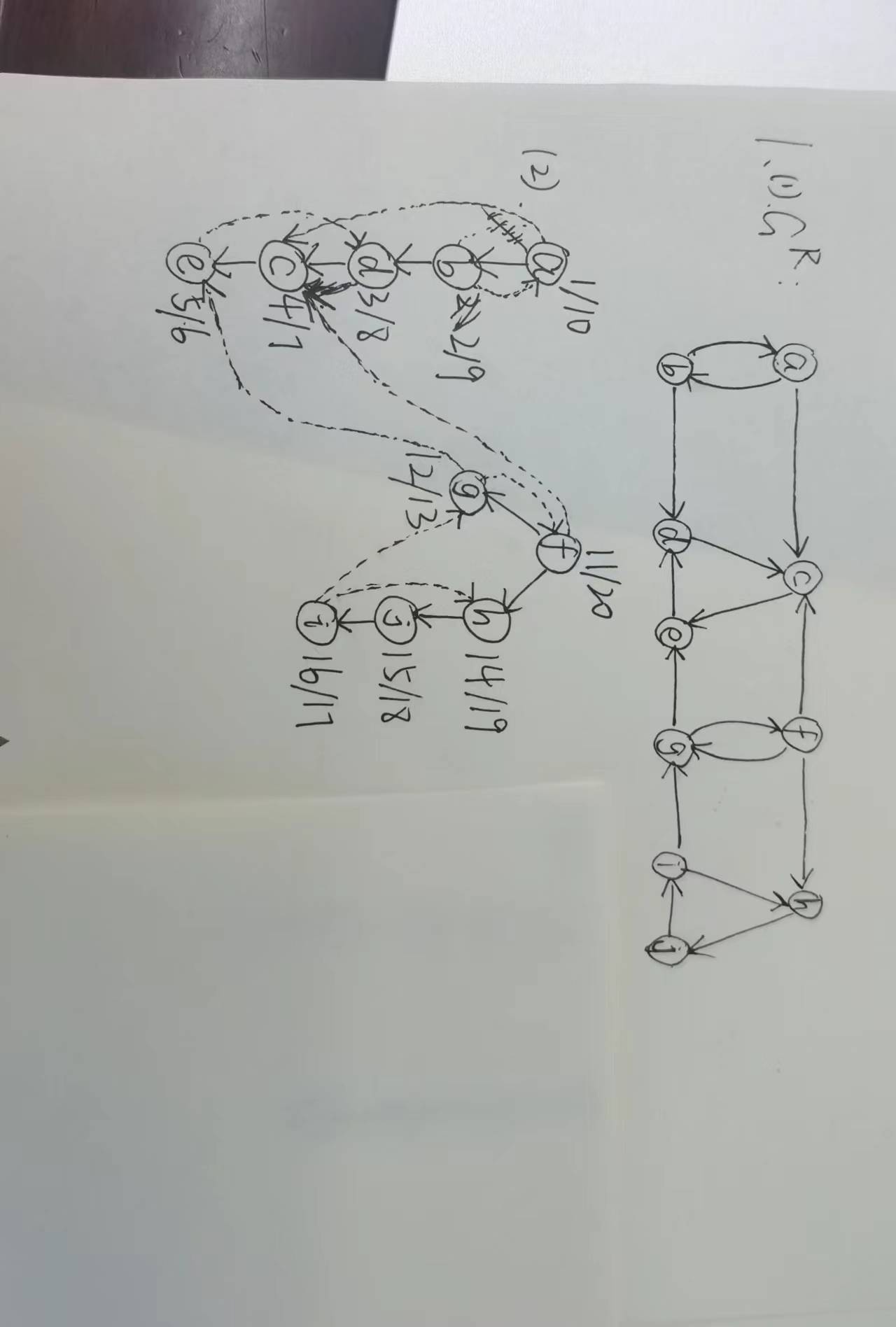
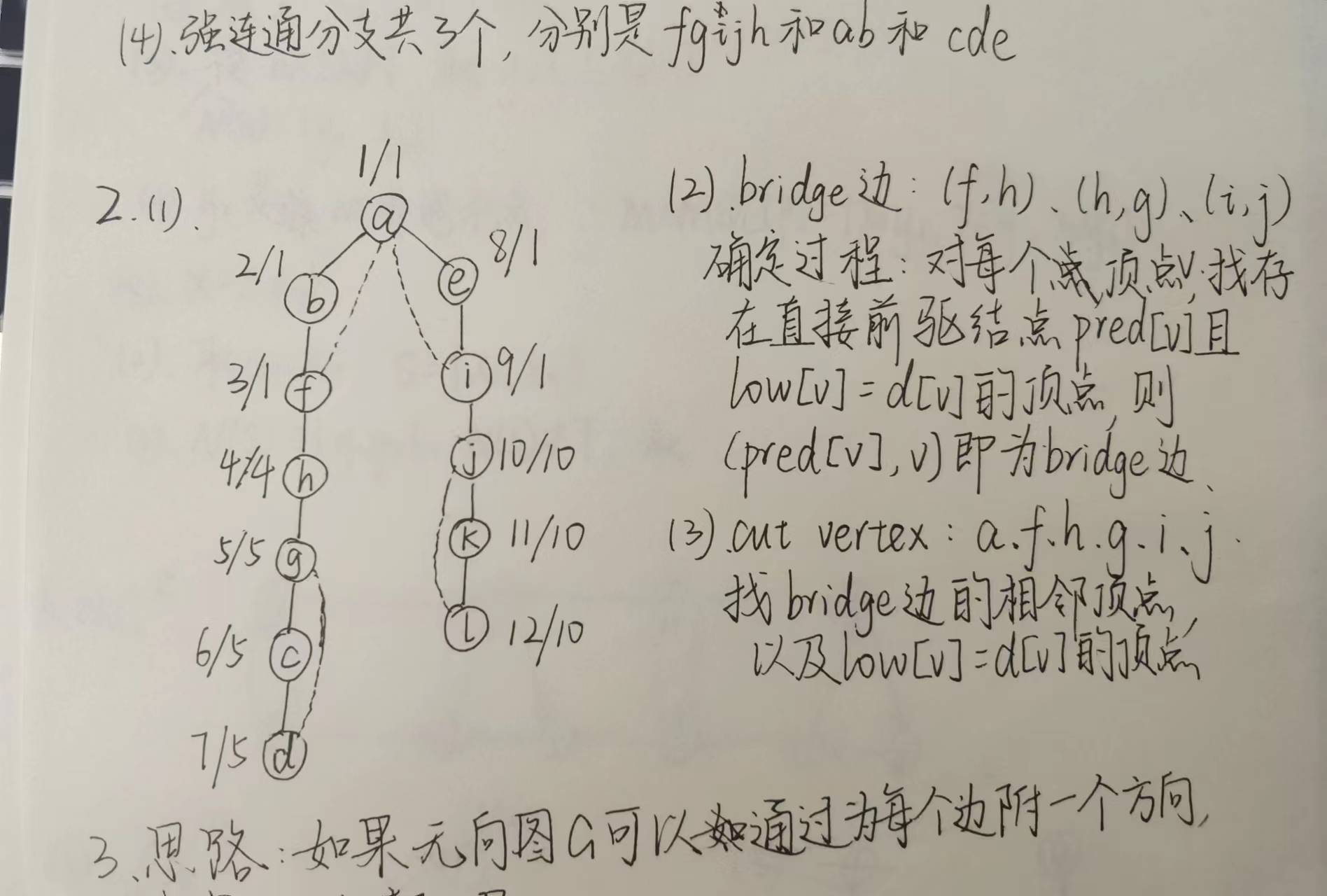
1.



2.



3.算法思路：如果无向图G可以通过为每一个边附一个方向，使得所有点都不是source点，则图G必含有环（回路），且所有顶点要么在环（回路）上，要么与环（回路）相连通。即只需要图G的每个连通分支都包含环（回路）即可。

因此，在DFS检测是否有环的算法基础上，对每个连通分支进行判断，只要有一个连通分支无环，图G就不符合要求，就可以停止遍历。只有当所有连通分支遍历完，均包含环时，才能判定图G满足要求

DFS(G) {

time = 0

for each (u in V)

mark[u] = undiscovered

for each (u in V) {

int flag = 0;//标记是否有环，有则为1；无则为0

if (mark[u] == undiscovered)

DFSVisit(u, &flag)

if( ! flag){ //只要任意一个连通分支无环，则图G不满足要求

printf(“No\n”);

return;

}

}

printf(“Yes\n”);//遍历完所有连通分支，均有环，则图G满足要求

}

DFSvisit(u, int \* flag) {

mark[u] = discovered

d[u] = ++time

for each (v in Adj(u)) {

if (mark[v] == undiscovered) {

pred[v] = u

DFSvisit(v, flag)

} else if (mark[v] != finished && v != pred[u]) {

\*flag = 1; //有环，将flag置为1

}

}

mark[u] = finished

f[u] = ++time

}