

```

queue <
P, vector <
P>
, greater <
P>>
q; memset(d, inf, sizeof(d)); d[s] =
0; q.push(P(0, s)); while(!q.empty()) Pp = q.top(); q.pop(); intu = p.second; if(d[u] < p.first) continue; for(inti = 0; i < g
ford(ints) memset(d, inf, sizeof(d)); d[s] = 0; //n - 1, nfor(inti = 1; i <= n; i++) intflag = 0; for(intj = 0; j < edgect
ack(Edgev, w); a[v].push_back(Edgeu, w);
queue <
P, vector <
P>
, greater <
P>>
que; //ms(dist, INF); //ms(dist2, INF); fill(dist, dist+
n, INF); fill(dist2, dist2+
n, INF); dist[0] =
0; que.push(P(0, 0)); while(que.size()) Pu = que.top(); que.pop(); intv = u.second; lld = u.first; if(dist2[v] < d) continue;
jintdep[maxn], n, m, rt, fa[maxn][20]; intv[maxn] =
0; //
ack(b); son[b].push_back(a); memset(fa, 0, sizeof(fa)); memset(dep, inf, sizeof(dep)); v[0] =
1; dep[0] =
0; dfs(0, rt); for(inti =
1; i <=
m; i+
+)inta, b; scanf("printf("return0;
1;
c; //nowDFSdfs_cdep[dfs_c] =
d; //nowvis[dfs_c+
+] =
now; //DFSdfs_cnowdfs_cfor(inti =
0; i <
g[now].size(); i+
+)if(g[now][i] != pre)dfs(now, g[now][i], d + 1); vis[dfs_c] = now; dep[dfs_c++] = d;
c; i+
+)cout <<
dep[i] <<
""; cout <<
""; for(inti =
1; i <=
dfs_c; i+
+)cout <<
vis[i] <<
""; cout <<
""; for(inti =
1; i <=
n; i+
+)cout <<
id[i] <<
""; cout <<
"";
ack(b); g[b].push_back(a); dfs(0, rt, 1); getSt(dfs_c); //check(n); for(inti =
1; i <=
m; i+
+)inta, b; scanf("printf("return0;
1].iscut =
1; elselow[u] = min(low[u], dfn[v]);
1].val+ =
a; returna; return0;
1].val+ =
a; rflow- =
a; //if(rflow <=
0)break; //break//udepi f(rflow ==
flow)dep[u] =
-2; returnflow-
rflow; //
ack(); if(find(temp.node1) !=
find(temp.node2))sum+ = temp.cost; n--; join(temp.node1, temp.node2); if(n! =
ledges.empty())sum =
-1; returnsum;
ack(t); result =
kruskal(n); if(result ==
-1)cout <<
"orz" <<
endl; elsecout <<
result <<
endl; return0;
queue <
P, vector <
P>
, greater <

```