LTDT

void breath\_first\_search(Graph\* G, int start) {

Queue frontier;

make\_null\_queue(&frontier);

push(&frontier, start);

mark[start] = 1;

while (!empty(&frontier)) {

int x = top(&frontier);

pop(&frontier);

printf("%d\n", x);

List list = neighbors(G, x);

int j;

for (j = 1; j <= list.size; ++j) {

int y = element\_at(&list, j);

if (!mark[y]) {

mark[y] = 1;

push(&frontier, y);

}

}

}

}

int main() {

//freopen("dt.txt", "r", stdin);

Graph G;

int n, m, i, j, x, y;

scanf("%d%d", &n, &m);

init\_graph(&G, n);

for (i = 1; i <= m; ++i) {

scanf("%d%d", &x, &y);

add\_egde(&G, x, y);

}

for (i = 1; i <= n; ++i) {

mark[i] = 0;

}

for (i = 1; i <= n; ++i) {

if (!mark[i]) {

breath\_first\_search(&G, i);

}

}

return 0;

}