



# Maximal Tourism

locked

H by philipsweng

Problem

Submissions

Leaderboard

Discussions

Editorial

H Editorial by philipsweng

One possible way to solve this problem is to iterate every location and let it be the start place, do a dfs and find out how many different locations it can reach. An crucial notice is that for the buses are all bilateral, if a location has been visited, we can just ignore it because the answer will not change.

## Statistics

Difficulty: Medium

Time Complexity:  $O(n + m)$ 

Required Knowledge: graph, dfs

Publish Date: Apr 10 2017

H Set by philipsweng

Problem Setter's code :

## include

```
using namespace std;

const int maxn = 100005;

vector lk[maxn]; bool walk[maxn]; int n,m,ans,cnt;

void dfs(int now) { if (walk[now]) return; ++ cnt; walk[now] = 1; for(auto p : lk[now]) dfs(p); }

int main() { scanf("%d%d", &n, &m); for(int i = 1,u,v;i <= m;i++) { scanf("%d%d", &u, &v);
lk[u].push_back(v),lk[v].push_back(u); } for(int i = 1;i <= n;i++) if (!walk[i]) { cnt = 0; dfs(i); ans =
max(ans,cnt); } printf("%d\n", ans); return 0; }
```

Join us on IRC at [#hackerrank](#) on freenode for hugs or bugs.[Contest Calendar](#) | [Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)