```cpp
//
//   Given DAG, find the length of the longest path in G.
//
//   Time Complexity: O(N)
//

#include <bits/stdc++.h>

using namespace std;

vector<vector<int>> adj;
vector<int> dp;

int dfs(int node) {
    if (dp[node] != -1) return dp[node];

    int longest = 0;
    for (int i = 0; i < adj[node].size(); i++) {
        longest = max(longest, dfs(adj[node][i]) + 1);
    }

    dp[node] = longest;
    return longest;
}

int main() {
    int n, m;
    cin >> n >> m;

    adj.resize(n);
    dp.resize(n, -1);

    for (int i = 0; i < m; i++) {
        int x, y; cin >> x >> y; x--; y--;
        adj[y].push_back(x);
    }

    int ans = 0;

    // DFS from each vertex
    for (int i = 0; i < n; i++) {
        ans = max(ans, dfs(i));
    }

    cout << ans << endl;
    return 0;
}
```