```
#include <iostream>
#include <vector>
#include <climits>

void testcase() {
    int num_positions, num_transitions;
    std::cin >> num_positions >> num_transitions;
    int red_start, black_start;
    std::cin >> red_start >> black_start;
    std::vector<std::vector<int> > transitions(num_positions, std::vector<int>
());
    for(int i = 0; i < num_transitions; i++) {
        int from, to;
        std::cin >> from >> to;
        transitions.at(from).push_back(to);
    }

    std::vector<int> next_step_shortest(num_positions, INT_MAX);
    std::vector<int> next_step_longest(num_positions, 0);
    for(int position = num_positions - 1; position > 0; position--) {
        std::vector<int> reachable_positions = transitions.at(position);
        for(int reachable_position : reachable_positions) {
            if(reachable_position == num_positions) {
                next_step_shortest.at(position) =
std::min(next_step_shortest.at(position), 1);
                next_step_longest.at(position) =
std::max(next_step_longest.at(position), 1);
            }
            else {
                next_step_shortest.at(position) =
std::min(next_step_shortest.at(position),
next_step_longest.at(reachable_position) + 1);
                next_step_longest.at(position) =
std::max(next_step_longest.at(position),
next_step_shortest.at(reachable_position) + 1);
            }
        }
    }
    int winner;
    if(next_step_shortest.at(red_start) < next_step_shortest.at(black_start)) {
        // Holmes wins
        winner = 0;
    } else if(next_step_shortest.at(red_start) >
next_step_shortest.at(black_start)) {
        // Moriatry wins
        winner = 1;
    } else if(next_step_shortest.at(red_start) % 2 == 0){
        // Moriatry wins: Moriatry will be faster if number required steps is
odd
        winner = 1;
    } else {
```

```
        winner = 0;
    }
    std::cout << winner << std::endl;
}

int main() {
    std::ios_base::sync_with_stdio(false); // Always!
    int t; std::cin >> t;
    for(int i = 0; i < t; i++) {
        testcase();
    }
    return 0;
}
```