5 ビ太郎の旅 3 (Bitaro's Travel 3)

Author : 田中 優希 (blackyuki)

小課題 1

この小課題において，与えられるグラフはパスである．

頂点が $P_1, P_2, \ldots, P_N$ の順に並んでいるとする．

実装においては，次数が 1 の頂点が 2 つあるのでその片方に着目し，1 つずつ隣り合う頂点を辿っていくことで，P を復元することができる．

$s = P_i$ のとき，$v = (P_i, P_{i+1}, \ldots, P_j)$ が条件を満たすような最大の j を R_i とし，$v = (P_i, P_{i-1}, \ldots, P_j)$ が条件を満たすような最小の j を L_i とする．すると，P_i から出発して訪れることのできる街は $P_{L_i}, P_{L_i+1}, \ldots, P_{R_i}$ であり，答えは $N - (R_i - L_i + 1)$ となる．

L_i, R_i の値はそれぞれ愚直に $O(N)$ で求めることができる．全体の計算量は $O(N^2)$ であり，この小課題を解くには十分である．

小課題 2

この小課題において与えられるグラフは一般のグラフであるが，各 s について $O(N + M)$ 時間で答えを求めることができればよい．

各街 i に対し，(i, odd) と (i, even) の 2 つの仮想的な頂点を考える．

(i, odd) は $v_1 = s, v_l = i$ となるような奇数長の $v = (v_1, v_2, \ldots, v_l)$ が存在することを表し，(i, even) は $v_1 = s, v_l = i$ となるような偶数長の $v = (v_1, v_2, \ldots, v_l)$ が存在することを表す．辺 (i, j) $(i < j)$ が元のグラフに存在するならば，仮想グラフにおいては (i, odd) から (j, even) へ向かう有向辺と，(j, even) から (i, odd) へ向かう有向辺を張る．この仮想的なグラフにおいて，(s, odd) から (i, odd) もしくは (i, even) へ到達可能である時，またその時に限り，元のグラフにおいて，街 s から出発して街 i を訪れることができる．

この仮想グラフは頂点数 $2N$ で辺数 $2M$ であるため，幅優先探索や深さ優先探索を用いて (s, odd) から各頂点への到達可能性を $O(N + M)$ 時間で求めることができる．

全体の計算量は $O(N(N + M))$ となる．

小課題 3

この小課題では，小課題 1 の解法を高速化する必要がある．具体的には，L_i, R_i をそれぞれ $O(1)$ 時間で求める方法を示す．

$i = 1, 2, \ldots, N$ の順で L_i を求めていき，既に求めた L_i を再利用することを考える．$i \geqq 3$ かつ $P_{i-1} > P_i$ かつ $P_{i-2} < P_{i-1}$ のとき，$L_i = L_{i-2}$ となる．そうでなく，$i \geqq 2$ かつ $P_{i-1} > P_i$ のとき，$L_i = i - 1$ となる．そうでないとき，$L_i = i$ となる．

同様に，$i = N, N-1, \ldots, 1$ の順で R_i を求めることができる．

よって，全体の計算量は $O(N)$ となる．

小課題 4

この小課題では，各頂点の次数が高々 2 であることを利用する．グラフの連結成分は全てパスかサイクルとなる．この問題は連結成分ごとに独立して解くことができる．

パスの連結成分については，小課題 3 の解法をそのまま用いることができる．

サイクルの連結成分については，連結成分のサイズを n とし，$P_1, P_2, \ldots, P_n$ の順に頂点が並んでいるとする．すなわち，辺は $(P_1, P_2), (P_2, P_3), \ldots, (P_{n-1}, P_n), (P_n, P_1)$ である．ここで，$P_1, P_2, \ldots, P_n, P_1, P_2, \ldots, P_n$ と頂点を $2n$ 個並べたパスグラフを考える．このパスグラフに対して，小課題 3 と同様にして，各 $i = 1, 2, \ldots, 2n$ について L_i, R_i を求める．

すると，元のサイクルにおいて，P_i から出発して全ての頂点 $P_1, P_2, \ldots, P_n$ を訪れることができるならば $L_{i+n} = 1, R_i = 2n$ となる．そうでないとき，$v = (P_i, P_{i+1}, \ldots)$ と辿った時に訪れることのできる街の数は $R_i - i + 1$ であり，$v = (P_i, P_{i-1}, \ldots)$ と辿った時に訪れることのできる街の数は $i + n - L_{i+n} + 1$ である．P_i での重複を除くと，訪れることのできる街の数は $(R_i - i + 1) + (i + n - L_{i+n} + 1) - 1 = R_i - L_{i+n} + n + 1$ となる．よって，訪れることのできる街の数は $\min(n, R_i - L_{i+n} + n + 1)$ と表すことができる．

全体の計算量は $O(N)$ となる．

満点

この小課題では，小課題 2 の解法を高速化する必要がある．

小課題 2 の解法で構築した仮想グラフにおいて，(i, odd) から (j, even) へ向かう有向辺が存在することと，(j, even) と (i, odd) へ向かう有向辺が存在することは等価である．したがって，この仮想グラフは無向グラフであるため，幅優先探索や深さ優先探索，あるいは Union-find などを用いてあらかじめ連結成分ごとに分けておき，それぞれの連結成分について，(i, odd) と (i, even) のいずれかが含まれるような i の個数を記録しておくことができる．この値が，この連結成分に (s, odd) が含まれるようなすべての s に対し，街 s から

出発して訪れることのできる街の数である．

全体の計算量は $O(N + M)$ や $O((N + M)\alpha(N))$ となる．