

Sep 06, 12 20:35

verify\_path\_in\_tournament.cpp

Page 1/1

```

/* FILE: verify_path_in_tournament.cpp   last change: 8-Sept-2012   author: Romeo
o Rizzi
* a verifier for problem 1 (tournament) in the 25-06-2012 exam in Algorithms
* Usage syntax:
*   > verify_path_in_tournament <path_file> <tournament_file>
*/

#include <iostream>
#include <fstream>
#include <cassert>

using namespace std;

const int MAX_N = 2000;
int n; // Numero giocatori (numero di nodi nel tournament, un tournament e' un
grafo diretto completo).

// Matrice vittorie/sconfitte: T[i][j] = true se i batte j
bool T[MAX_N + 1][MAX_N + 1]; // i nodi sono numerati da 1 ad n.

int H[MAX_N + 1]; // il cammino hamiltoniano H dato come sequenza di nodi.
int n_H; // numero nodi nel cammino H

int main(int argc, char** argv) {
    ifstream fin_path(argv[1]); assert( fin_path );
    ifstream fin_tournament(argv[2]); assert( fin_tournament );
    fin_tournament >> n; fin_path >> n_H;
    if( n_H > n ) { cout << "il path ha piu' nodi del tournament." << endl; return 1; }
    if( n_H < n ) { cout << "il path ha meno nodi del tournament." << endl; return 2; }
    for(int i = 1; i <= n; i++)
        for(int j = 1; j <= n; j++)
            fin_tournament >> T[i][j];
    fin_tournament.close();

    for(int edge = 1; edge < n_H; edge++) {
        int node_from, node_to; fin_path >> node_from >> node_to;
        H[edge+1] = node_to;
        if( edge == 1 ) H[edge] = node_from;
        if( H[edge] != node_from ) { cout << "il path fa brutti salti." << endl; return 3; }
    }
    fin_path.close();

    for(int i = 1; i < n; i++)
        if(!T[ H[i] ][ H[i+1] ]) {
            cout << "the arc (" << H[i] << ", " << H[i+1] << ") is in the path but not in the tournament"
<< endl;
            return 4;
        }
    return 0;
}

```