



CIENCIAS DE LA
COMPUTACIÓN

CIRCUITO
HAMILTONIANO

ANÁLISIS Y DISEÑO DE
ALGORITMOS

Villanueva Borda, Harold

Alejandro

5º SEMESTRE

2022

“Los alumnos declaran haber realizado el presente trabajo de acuerdo
a las normas de la Universidad Católica San Pablo”

```

#include <iostream>
#include <fstream>

using namespace std;

template<class T>
T** create_matrix(int rows, int columns) {
    T** matrix = new T*[rows];

    for(int i = 0; i < rows; ++i) {
        *(matrix + i) = new T[columns];

        for(int j = 0; j < columns; ++j) {
            (*(matrix + i) + j ) = 0;
        }
    }
    return matrix;
}

template<class T>
void print(T** matrix, int rows, int columns) {

    for(int i = 0; i < rows; ++i) {
        for(int j = 0; j < columns; ++j) {
            cout << (*(matrix + i) + j ) << "\t";
        }
        cout<<endl;
    }
}

void printSolution(int *path, int m){
    cout << "si tiene circuito hamiltoniano" << endl;
    for (int i = 0; i < m; ++i)
        cout << *(path+i) << "\t";
    cout << path[0] << "\n\n";
}

template<class T>
T utility(int vertex, T **adjMatrix, const int *path, int position){
    if (adjMatrix [ path[position - 1] ][ vertex ] == 0)
        return false;

    for (int i = 0; i < position; ++i)
        if (path[i] == vertex)
            return false;

    return true;
}

template<class T>
bool util_hamilton(T **adjMatrix, int *path, int position, int m)
{
    if (position == m){
        if (adjMatrix[ path[position - 1] ][ path[0] ] == 1 )
            return true;
        else
            return false;
    }

    for (int v = 1; v < m; ++v)
    {
        if (utility(v, adjMatrix, path, position))
        {
            path[position] = v;

            if (util_hamilton(adjMatrix, path, position + 1, m) == true)
                return true;
            path[position] = -1;
        }
    }
}

```

```

    }
}

return false;
}

template<class T>
T hamiltonian_circuit(T **adjMatrix, int m){
    int *path = new int[m];
    for (int i = 0; i < m; i++)
        *(path+i) = -1;

    path[0] = 0;
    if (util_hamilton(adjMatrix, path, 1, m) == false ) {
        cout << "no existe circuito hamiltoniano" << endl;
        return false;
    }

    printSolution(path, m);
    return true;
}

int main(){

    bool **A, **B, **C, **D;
    int m = 5, n=8, o = 4;

    ifstream matrixA, matrixB, matrixC, matrixD;
    matrixA.open("C:/Users/win 10/Documents/UCSP/V CICLO/ADA/TAREA
3/matrix1.txt");
    matrixB.open("C:/Users/win 10/Documents/UCSP/V CICLO/ADA/TAREA
3/matrix2.txt");
    matrixC.open("C:/Users/win 10/Documents/UCSP/V CICLO/ADA/TAREA
3/matrix3.txt");
    matrixD.open("C:/Users/win 10/Documents/UCSP/V CICLO/ADA/TAREA
3/matrix4.txt");

    if(!matrixA.is_open() and !matrixB.is_open() and !matrixC.is_open() and
!matrixD.is_open()){
        cout << "Error" << endl;
        return 1;
    }

    A = create_matrix<bool>(m, m);
    B = create_matrix<bool>(m, m);
    C = create_matrix<bool>(n, n);
    D = create_matrix<bool>(n, n);

    for(int i = 0; i < o; ++i)
        for(int j = 0; j < o; ++j)
            matrixA >> (*(A + i) + j );

    for(int i = 0; i < m; ++i)
        for(int j = 0; j < m; ++j)
            matrixB >> (*(B + i) + j );

    for(int i = 0; i < m; ++i)
        for(int j = 0; j < m; ++j)
            matrixC >> (*(C + i) + j );

    for(int i = 0; i < n; ++i)
        for(int j = 0; j < n; ++j)
            matrixD >> (*(D + i) + j );

    cout << endl;
    print<bool>(A, o, o);
    cout << endl;
    hamiltonian_circuit(A, o);
}

```

```
cout << endl;
print<bool>(B, m, m);
cout << endl;
hamiltonian_circuit(B, m);

cout << endl;
print<bool>(C, m, m);
cout << endl;
hamiltonian_circuit(C, m);

cout << endl;
print<bool>(D, n, n);
cout << endl;
hamiltonian_circuit(D, n);

matrixA.close();
matrixB.close();
matrixC.close();
matrixD.close();

for (int i = 0; i < o; ++i)
    delete[] A[i];
delete[] A;

for (int i = 0; i < m; ++i)
    delete[] B[i];
delete[] B;

for (int i = 0; i < m; ++i)
    delete[] C[i];
delete[] C;

for (int i = 0; i < o; ++i)
    delete[] D[i];
delete[] D;

}
```

0	0	0	1
1	0	0	1
1	1	0	0
1	0	1	0

si tiene circuito hamiltoniano

0	3	2	1	0
---	---	---	---	---

0	1	0	1	0
1	0	1	1	1
0	1	0	0	1
1	1	0	0	1
0	1	1	1	0

si tiene circuito hamiltoniano

0	1	2	4	3	0
---	---	---	---	---	---

0	1	0	1	0
1	0	1	1	1
0	1	0	0	1
1	1	0	0	0
0	1	1	0	0

no existe circuito hamiltoniano

0	1	1	0	0	0	0	0
1	0	1	1	0	0	1	0
1	1	0	1	0	0	0	1
0	1	1	0	1	0	0	1
0	0	0	1	0	1	1	1
0	0	0	0	1	0	1	0
0	1	0	0	1	1	0	1
0	1	1	1	1	0	1	0

si tiene circuito hamiltoniano

0	1	3	4	5	6	7	2	0
---	---	---	---	---	---	---	---	---