**UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS**

**FACULTAD DE INGENIERÍA DE SOFTWARE**



**Complejidad Algorítmica**

**Integrantes:**

Malca Rojas, Nicoll Yamira - U201820152

Brocca Hernandez Erick Alonso U201822206

**Profesor:** Canaval Sanchez, Luis Martín

**Ciclo: 2021-01**

Contenido

[Introducción 3](#_Toc71381182)

[Objetivos del proyecto 4](#_Toc71381183)

[Marco Teórico 4](#_Toc71381184)

[ Grafos 4](#_Toc71381185)

[ Backtracking: 5](#_Toc71381186)

[Análisis de la Complejidad Algorítmica 5](#_Toc71381187)

[Tiempo asintótico 6](#_Toc71381188)

[Conclusiones 6](#_Toc71381189)

[Tabla de ilustraciones 7](#_Toc71381190)

[Bibliografía 7](#_Toc71381191)

# Introducción

El presente trabajo correspondiente al curso de Complejidad Algorítmica tiene como objetivo que el estudiante aplique los conocimientos aprendidos hasta la fecha. El trabajo demanda hacer un buen uso de los respectivos paradigmas de programación que se adecuen al problema abordado. Puesto que, por su naturaleza, un problema tiene la capacidad de ser solucionado por uno o varios métodos, también se deberá evaluar la viabilidad de las soluciones posibles porque, ya que si bien es importante llegar a la respuesta, más importante es elegir la más conveniente.

El problema asignado al estudiante en este caso es el problema del agente viajero o más conocido por sus siglas en inglés como TSP.  Este es un problema algorítmico donde el principal objetivo de su solución es encontrar la ruta más corta entre un conjunto de puntos o locaciones que deben ser visitados, en este ya conocido problema cada punto representa una ciudad que un viajero podría visitar. El objetivo del viajero es mantener los costos de viaje y distancia lo más bajos que sean posible, de modo que para visitar un punto X desde un punto Y, se tendría que elegir el camino mas corto entre estos dos puntos tomando en consideración las distancias entre cada punto y el coste necesario para poder visitarlo.

Se solicita hacer la implementación de este problema en nuestro país tomando como puntos de visita los centros poblados de cada distrito pertenecientes a una provincia en específico. El equipo de trabajo está conformado por dos integrantes y el proceso de trabajo consta de avance semanales del mismo y reporte de cada uno de los miembros Asimismo se hará uso de diversos recursos disponibles como GitHub (el más importante), tutoriales, libros, guías, etc.

# Objetivos del proyecto

En este presente trabajo, tenemos como objetivos del proyecto lo siguiente:

* Investigar acerca de la técnica empleada en este proyecto, la cual es backtraking
* Implementar la técnica backtraking para solucionar el problema del vendedor viajero
* Analizar la complejidad algorítmica del algoritmo presentado.

# Marco Teórico

Para la solución de nuestro problema es necesario saber algunos conceptos para su aplicación:

## Grafos

El grafo es considerado como la estructura mas general dentro de la clasificación de estructura de datos, ya que esta se basa en la generalización de una estructura jerárquica

Se define a un grafo como una estructura de datos que nos permite la representación de varias relaciones entre objetos. El grafo consta de dos elementos que son los vértices y las aristas, estas ultimas sirven para conectar un vértice con otro, por ejemplo, la figura 1.0 representa en sus vértices diferentes caminos enumerados desde el 1 al 8, y las aristas de estos representan la distancia en km que existe entre cada punto.

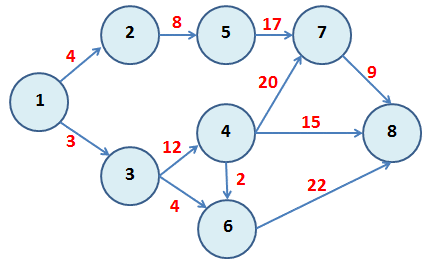


Ilustración 1: Ejemplo de grafo

## Backtracking:

El backtracking realiza una extensión de una solución parcial. En cada ciclo de búsqueda si una extensión de la solución actual no es posible, se regresa a una solución parcial corta y se vuelve a tratar de la misma manera. El método de backtracking es usando en un amplio escenario de problemas de búsqueda, este va desde implementación de parsers, desarrollo de videojuegos y calendarización, por este motivo este algoritmo es el que se implementó en nuestro trabajo. Otros ejemplos de problemas comunes resuelto utilizando este algoritmo serian el juego Sudoku, el problema de los movimientos de un caballo y el problema de las ocho reinas

# Análisis de la Complejidad Algorítmica

A continuación, se detallará la complejidad algorítmica de la estrategia usada para solucionar el problema, es decir backtracking.

Se implementó el siguiente código:

const TSP\_BACKTRACKING = (graph, length\_solution, recorded\_distance, result) => {

    if(length\_solution == graph.cities.length - 1)

    {

        result = Math.min(

            result,

            recorded\_distance +

            graph.getDistance(graph.cities[graph.cities.length - 1], graph.cities[0])

        )

    }

    else

    {

        for (let i = length\_solution + 1; i < graph.cities.length; i++)

        {

            graph.swapCities(length\_solution + 1, i)

            let new\_distance = recorded\_distance + graph.getDistance(graph.cities[length\_solution], graph.cities[length\_solution + 1]);

            if (new\_distance >= result)

            {

                continue;

            }

            else

            {

                result = Math.min(result, TSP\_BACKTRACKING(graph, length\_solution + 1, new\_distance, result));

            }

            graph.swapCities(length\_solution + 1, i)

        }

    }

    return result;

}

El tiempo asintótico total para este algoritmo es:

**O(2^n)**

## Tiempo asintótico

Ilustración 2: Tiempo asintótico del Código

# Conclusiones

Como conclusión de nuestro informe del trabajo parcial cabe mencionar que:

-Se ha demostrado la capacidad para la interpretación, comunicación y uso de información diversa para la solución en una situación de contexto real

-El algoritmo backtracking es muy versátil en su uso y muy intuitivo al momento de implementarlo y es muy importante su aprendizaje

-El uso de grafos, tiene una amplia aplicación no solo en ámbitos matemáticos sino también computacionales y por esto es muy importante su aprendizaje en el curso

**NOTA:**

* Es importante mencionar que, para la visualización del mapa en este proyecto, se necesita de un servidor. Es por ello por lo que nosotros lo abrimos con GitHub. Enlace para observar el trabajo: <https://nikol-0822.github.io/TSP/>

# Tabla de ilustraciones

[Ilustración 1: Ejemplo de grafo 4](file:///C:\Users\NICOLL\Downloads\Trabajo_Parcial_Complejidad.docx#_Toc71380065)

[Ilustración 2: Tiempo asintótico del Código 6](file:///C:\Users\NICOLL\Downloads\Trabajo_Parcial_Complejidad.docx#_Toc71380066)

# Bibliografía

* Brucato, C.(2013). The Travelling Salesman Problem. (Master of Sciences, University of Pittsburgh. United States, Pittsburgh). Consultado en: <https://www.mathematics.pitt.edu/sites/default/files/TSP.pdf>
* Backtracking / Branch-and-Bound. (2015). Consultado en: <http://www.win.tue.nl/~bjansen/courses/2ILC0/backtracking.pdf>