**Año del Bicentenario del Perú: 200 años de Independencia**

**UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS**

**Facultad de Ingeniería**



**Informe WASC – Complejidad Algorítmica**

**Profesor:**

Canval Sánchez, Luis Martin

**UPC, noviembre del 2021**

**Introducción:**

El propósito de este informe es detallar, mediante la explicación y desarrollo del trabajo final de Complejidad Algorítmica, la competencia general del curso ya mencionado. La competencia Razonamiento Cuantitativo es la capacidad de interpretar, representar, comunicar y utilizar información cuantitativa diversa en situaciones de contexto real. Incluye calcular, razonar, emitir juicios y tomar decisiones con base en esta información cuantitativa; nivel de logro 1 en Ingeniería de Software y 2 en Ciencias de la computación.

En el trabajo final de este curso, se nos pidió darle una solución al problema llamado “Vehicle Routing Problem”, el problema proponía buscar un algoritmo el cuál pueda indicar las rutas óptimas para que un camión de entregas pueda visitar almacenes y destinos de entrega de la manera más óptima posible.

**Desarrollo del problema**

Uno de los primeros pasos a tomar para buscarle solución a este problema, es interpretar los distintos algoritmos aprendidos a lo largo de las clases, al interpretarlos podríamos inferir que algoritmo era el indicado para resolver este problema de contexto real. Una vez teniendo los algoritmos y habiendo aprendido su funcionamiento y para qué eran útiles debíamos adaptarlo a nuestras representaciones del problema. En el caso de mi grupo, representamos los puntos de entrega y almacén de la misma manera, posición X & Y, claramente usando menos almacenes que punto de entrega.

La comunicación fue vital para el uso de nuestros algoritmos y, como la representación de nuestro problema va a afectar la implementación de un código en Python funcional para darle solución al VRP. Una de las primeras implementaciones fue con el algoritmo BFS, presentado en el Hito 2 de nuestro repositorio en Github, viendo la cantidad de puntos de entrega y almacenes, el algoritmo fue ejecutado en un tiempo de 20 minutos sin errores, lo único negativo fue el hecho de que no podía presentar los nodos recorridos en jupyter notebook.

La segunda implementación de un código en nuestro trabajo, fue de Dijkstra’s algorithm, recorrimos los almacenes y hallamos un camino óptimo del almacén a un punto de entrega. Esta vez, tuvimos éxito al poder imprimir los el camino óptimo de cada punto de entrega. El siguiente paso era aplicar un algoritmo el cual pueda recorrer de un punto de salida del camión a los almacenes para posteriormente ir hacia los puntos de entrega, una vez visitado todos los puntos de entrega, seguir con el siguiente almacén y así hasta poder recorrer toda la ciudad.

Cada uno de los integrantes tuvo ideas separadas para estos algoritmos dependiendo de cómo representaban el problema y lo que veían más óptimo para su solución, se pudo lograr la implementación correcta de cada código y fue subido al repositorio en github.

**Conclusión**

Este curso y proyecto final me ha dado una nueva perspectiva sobre cómo funcionan los algoritmos enseñados y cómo cada uno varía según su time complexity. También, logré profundizar sobre teoría de grafos y cómo trabajar en github de una manera satisfactoria. Cada programador debería tener conocimiento sobre estos conceptos debido a que diversas plataformas utilizan estas teorías para tener un excelente funcionamiento de sus start-ups.