

# Sistemas Inteligentes

## Laboratorio 1

### Algoritmos de Búsqueda: Coloreo de Grafos

Profesor: Elizabeth Montero

Ayudantes: Alexander Espina & Nicolás Hormazábal

*Universidad Andrés Bello, Facultad de Ingeniería*  
Santiago-Viña del Mar, Chile.

## 1. Objetivo

El objetivo de este Laboratorio es:

- Diseñar e implementar un algoritmo de búsqueda local para resolver instancias de diferente tamaño del problema de coloreo de grafos.
- Diseñar e implementar las operaciones necesarias para la implementación del algoritmo que resuelve el problema.
- Escribir un reporte de la evaluación de la calidad de las soluciones encontradas en las instancias de prueba entregadas.

## 2. Enunciado

Los problemas de coloreo de mapas (también conocido como coloreo de grafos) son problemas bien conocidos en las áreas de inteligencia artificial y optimización. Dado un conjunto de colores disponibles, el problema consiste en determinar de qué color pintar cada una de las regiones de modo que cada par de regiones adyacentes sean coloradas de diferente color. Así planteado, el problema corresponde a un problema de satisfacción de restricciones.

## 3. Acerca del Laboratorio

### 3.1. Reglas

- Se realiza en equipos de dos personas.
- Se entrega a través de *github classroom* a más tardar a las 23:59 h. del día especificado.
- Se debe entregar como un archivo con el nombre Informe.pdf en formato pdf y el código fuente de la implementación. Los entregables se detallan en la sección 3.2.
- Se debe incluir un archivo con el nombre integrantes.txt donde con dos líneas donde en cada una irá el RUN (con puntos y guión) seguido de su nombre completo, separado por un único carácter de espacio.
- El espacio para que uno de los dos integrantes registre su entrega es <https://classroom.github.com/a/c36j7k9H>
- No se revisarán entregas atrasadas.

### 3.2. Entregables

- Informe de dos páginas en formato pdf que incluya:
  - Breve descripción del algoritmo
  - Pseudo-código o diagrama de flujo del funcionamiento general del algoritmo.
  - Detalle de sus principales componentes.
  - Evaluación preliminar de resultados.
- Código fuente de la propuesta de resolución
  - Codificada en C++
  - Compilable en entorno Linux
  - Debe incluir un Makefile que permita compilar y ejecutar.
  - Entrada/salida:
    - Debe recibir por línea de comandos tres argumentos: nombre del archivo de entrada, nombre del archivo de salida y tiempo máximo de ejecución. El tiempo en segundos.
    - Debe mostrar por pantalla la mejor solución encontrada (cantidad de restricciones insatisfechas y detalle de color de cada uno de los nodos) tal como se muestra en la figura 1. La figura 1 muestra un ejemplo del formato entrada/salida solicitado. En el ejemplo se resuelve la instancia del problema descrita en el archivo ejemplo1.col ubicado en el directorio Instancias/ y salida

debe estar en un directorio Salidas/ con el mismo nombre de entrada, pero con la extensión *sal* en lugar de *col*. Se busca la mejor solución considerando 2 segundos de tiempo máximo de ejecución. La solución obtenida por *solver* no satisface una restricción y requiere colorear los dos primeros nodos del color 1, el tercer nodo color 3, el cuarto nodo de color 1 y el quinto de color 2.

- El formato de archivos de entrada a considerar se revisará en sesiones de Laboratorio del curso.

```
elizabeth@cancun:~\$ ./solver -i Instancias/ejemplo1.col
-o Salidas/ejemplo1.sal -t 2
```

**Figura 1:** Ejemplo archivo entrada

```
1
1 1 3 1 2
```

**Figura 2:** Ejemplo archivo salida, cada nodo está ordenado en función de su aparición, en la primera columna y los restantes, por su aparición en la segunda columna, para la columna *e* de los arcos, desde arriba hacia abajo

### 3.3. Fechas

- Fecha de entrega: Jueves 24 de Septiembre de 2020.
- No se reciben entregas atrasadas.

### 3.4. Evaluación

- La nota del Laboratorio considera el cumplimiento de un conjunto de objetivos y el Ranking de calidad de resultados tal como se muestra en la ecuación (1).

$$\text{Nota Tarea} = 6 \cdot \text{Cumplimiento\_Objetivos} + \text{Ranking} \quad (1)$$

donde Cumplimiento\_Objeticos y Ranking varían entre 0 y 1.

- El cumplimiento de objetivos considera:
  - Implementación: Código correctamente implementado, separado en funciones, correctamente comentado y sin problemas de compilación ni ejecución.
  - Cumplimiento reglas: Informe y Makefile.
  - Explicación: Detalle de la propuesta en el Informe.
  - Justificación: Justificación de la elección y uso de heurísticas en la resolución.

- Evaluación: Ejecución, exposición y comentarios de resultados en las instancias propuestas considerando calidad de soluciones y tiempos de ejecución.
- El Ranking se calcula de acuerdo al desempeño de todas las propuestas entregadas. Para calcular el ranking de todas las propuestas:
  - Se seleccionarán tres instancias de diferente tamaño.
  - Se fijará un tiempo de ejecución máximo para cada una de ellas.
  - Se ejecutará cada algoritmo cinco veces. El mejor de esos cinco resultados se utilizará para calcular el ranking del equipo correspondiente.
  - La suma de los rankings ( $SR_i$ ) respecto a cada instancia se utiliza para calcular su influencia en la nota del equipo.

$$\text{Ranking}_i = \frac{SR_i}{\max_j \{SR_j\}} \quad (2)$$