

Trabajo Práctico Especial

Estructuras de Datos y Algoritmos

Primer Cuatrimestre 2010

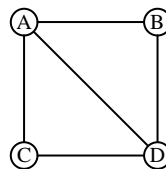
Objetivo

El objetivo del trabajo práctico es desarrollar una aplicación para realizar coloreo mínimo de grafos, obteniendo soluciones exactas y aproximadas a través de diferentes métodos de búsqueda.

Requerimientos

La aplicación deberá tomar como entrada un archivo con la estructura de un grafo, y realizar un coloreo en base al algoritmo elegido por el usuario. El archivo debe tener extensión “.graph”, y almacena en cada línea una arista del grafo, en donde se detalla el nodo origen y el nodo destino separados por coma. Los nombres de los nodos pueden ser cualquier secuencia de caracteres alfanuméricos. A continuación se muestra un archivo de ejemplo con el grafo correspondiente que representa:

```
A,B
A,D
A,C
B,D
C,D
```



Como salida el programa debe generar un archivo con el mismo nombre que el archivo de entrada, y extensión “.dot” con el grafo coloreado, para poder ser graficado por la herramienta Graphviz. Además, se debe generar un archivo con el mismo nombre que el de entrada y extensión “.color” en donde se listan todos los nodos del grafo y el color asignado a los mismos (número entero comenzando desde 0). Para el grafo anterior, una posible salida sería:

```
A=0
B=1
C=1
D=2
```

Los métodos que se deben implementar son:

- exact: debe encontrar una solución exacta al problema
- ts: debe encontrar una solución aproximada aceptable, utilizando la estrategia “*Tabu search*”
- greedy: debe encontrar una solución aproximada utilizando una estrategia “*Greedy*”

Además, si se eligió la solución exacta, el usuario puede opcionalmente solicitar que se genere un archivo adicional llamado “tree.dot” con el árbol de búsqueda utilizado por el algoritmo de coloreo para explorar el espacio de estados. El árbol debe mostrar todos los estados que fueron analizados por el programa.

La sintaxis para el uso de la aplicación es la siguiente:

```
# java -jar tpe.jar archivo metodo [tree]
```

Ejemplos de uso:

```
# java -jar tpe.jar prueba.graph exact tree
```

Genera los archivos prueba.colors y prueba.dot con el coloreo mínimo del grafo representado en el archivo prueba.graph, utilizando un algoritmo que implemente la solución exacta, y además el archivo tree.dot con el árbol de búsqueda utilizado.

```
# java -jar tpe.jar prueba.graph ts
```

Genera los archivos prueba.colors, prueba.dot con el coloreo mínimo del grafo representado en el archivo prueba.graph usando la heurística Tabu Search, y no genera el árbol de búsqueda.

Material a entregar

Cada grupo deberá dejar en el repositorio el siguiente material

1. Códigos fuentes. Se debe incluir un *buildfile* de Ant que permita generar el archivo *jar* de la aplicación.
2. Archivos de grafos usados para las pruebas
3. Un documento, en formato pdf, que explique en forma clara:
 - Problemas reales que se pueden solucionar con coloreo de grafos
 - Algoritmos creados tanto para la solución exacta como aproximada
 - Cálculo de complejidad temporal y espacial para cada algoritmo
 - Tablas de comparación de tiempos para cada algoritmo
 - Conclusiones

Fecha de entrega

El viernes 28 de mayo, antes de las 18 hs cada grupo deberá enviar un mail a la cuenta de la materia indicando el número de versión a considerar.

El lunes 31 de mayo, al comienzo de la clase de Taller, cada grupo deberá presentar impreso el documento mencionado en el punto 3 del **Material a entregar**.