

Análisis y Diseño de Algoritmos II

Trabajo Práctico N°2

Implemente en C++ la consigna asignada (ver GruposTPEAyDA2020.pdf)

1. **Problema del coloreo:** Un coloreo de un grafo $G=(V,A)$ es una transformación $C: V \rightarrow S$ donde S es un conjunto finito de colores tal que si (v,w) pertenece a A , luego: $C(v) \neq C(w)$. Es decir, vértices adyacentes no pueden tener el mismo color. El número cromático de G , llamado $X(G)$ es el mínimo número de colores necesarios para colorear G . Escriba un algoritmo que encuentre $X(G)$ e imprima el grafo poniendo al lado del número de vértice el color con el que está pintado. Resuelva por Backtracking.
2. **Tablero mágico.** Dado un tablero de tamaño $n \times n$, construir un algoritmo que ubique (si es posible) $n \times n$ números naturales diferentes, entre 1 y un cierto k , de manera tal que la suma de las columnas y de las filas sea S . Resuelva por backtracking.
3. **Variante del problema de las n reinas:** Supongamos que las casillas del tablero contienen números naturales entre 1 y un m dado. Se requiere encontrar todas las asignaciones de n reinas al tablero de forma tal que no se capturen entre sí y que la suma de los valores de las posiciones del tablero en las que se ubicaron sea menor o igual que un valor k . Resuelva por backtracking.
4. **Problema de la asignación de tareas:** Se tienen que asignar n tareas a n personas, de forma que cada persona realice exactamente una tarea. Si a la persona i se le asigna la tarea j , entonces el costo de realizarla será C_{ij} . Dada una matriz de costos C , el problema consiste en asignar todas las tareas de forma tal que se minimice el costo total de ejecutar las n tareas. Resuelva por Backtracking.
5. **Laberinto.** Dado un laberinto consistente en una matriz cuadrada que tiene en cada posición un valor natural y cuatro valores booleanos, indicando estos últimos si desde esa casilla se puede ir al norte, este, sur y oeste, encontrar un camino de longitud mínima entre dos casillas dadas, siendo la longitud de un camino la suma de los valores naturales de las casillas por las que pasa. Resuelva por backtracking.
6. **Problema de la mochila – versión 2:** Dados n objetos cada uno de los cuales tiene asociado un peso p_i y un beneficio b_i con $1 \leq i \leq n$ y una mochila de

capacidad C , encontrar un subconjunto de objetos que maximice el beneficio con la restricción la suma de los objetos no supere la capacidad de la mochila y los objetos no se pueden fraccionar. Escriba un algoritmo que resuelva este problema:

- a. Backtracking.
- b. Algoritmo aproximado S_k

7. **Problema del cubrimiento de conjuntos (Set-Covering):** Supongamos tener M llaves y N cerraduras. Cada llave abre una o más cerraduras. Implemente un algoritmo para encontrar un conjunto mínimo de llaves que abran todas las cerraduras. Escriba un algoritmo que resuelva este problema:

- a. Backtracking
- b. Algoritmo de aproximación.

8. **Problema del vertex cover:** el vertex-cover de un grafo no orientado $G = (V, A)$ es un subconjunto $V' \subseteq V$ tal que, $\forall (u, v) \in A$ entonces $u \in V'$ o $v \in V'$ (o ambos). El tamaño del vertex cover es el número de vértices. Escriba un algoritmo para encontrar el vertex cover de mínimo tamaño:

- a. Backtracking
- b. Algoritmo de aproximación.

9. **Problema del viajante:** Un viajante de comercio debe visitar una serie de ciudades. Cada ciudad está conectada con las restantes mediante rutas que tienen un costo asociado. El problema consiste en hallar la ruta que deberá tomar para visitar todas las ciudades exactamente una vez retornando a la ciudad de partida, con el menor costo posible. Escriba un algoritmo que resuelva este problema:

- a. Backtracking
- b. Algoritmo de aproximación para el Problema del viajante métrico suponiendo que siempre es menos costoso ir directamente de una ciudad a otra que pasando por una ciudad intermedia, es decir, la función de costo satisface la desigualdad triangular: para toda ciudad u, v, w , $\text{costo}(u, w) \leq \text{costo}(u, v) + \text{costo}(v, w)$.

Aclaraciones:

1. La información que registra/devuelve el algoritmo deberá imprimirse por pantalla con un nivel de detalle suficiente para que se interprete claramente la respuesta del mismo.

2. Para la entrega deberán subir una carpeta comprimida a moodle (Tarea: TPE2-AyDA2020) con el proyecto (.cbp) y los códigos fuentes con la resolución y todo archivo que considere necesario para la corrección.
 - a. El nombre de la carpeta deberá respetar el siguiente formato: "TPE2-GX-CY.zip" donde X es el número del grupo e Y la consigna que resolvieron.
 - b. Solo uno de los integrantes del grupo tiene que subir el trabajo.
 - c. No se aceptarán entregas de grupos que hayan resuelto una consigna que no corresponde con la asignada.
3. La fecha límite para subir los trabajos es el lunes 8 de junio.