UNIVERSIDAD DEL SURESTE: DE LA FRONTERA “COMALAPA”.

ASIGNATURA: Algoritmos y estructuras de datos.

ALUMNO: Ramiro Gerardo Resendíz Valdéz

DOCENTE: Icel Bernardo Lepe Arriaga.

CARRERA: Ingeniería en sistemas computacionales (ISC).

CUATRIMESTRE: Quinto (5^{to}).

GRUPO: ISC13SDC0220-A.

PARCIAL: Primero (1^{ro}).

TRABAJO: Ensayo de la unidad uno de la antología.

FECHA DE ENTREGA: 05/Marzo/2022.

Diseño y análisis de algoritmos “conceptos básicos de algoritmos”.

No hay una regla rigurosa para redactar un programa que resuelva un dado problema tácticas para solucionar inconvenientes y equiparar a priori la eficiencia de las mismas. Por (Traveling Salesman Problem) el cual se apoya en hallar el orden en que se debería recorrer un cierto número de localidades (esto es, una secuencia de aspectos en el plano) en forma. Este problema nace en una pluralidad de aplicaciones prácticas, ejemplificando, descubrir senderos mínimos para recorridos de repartición de productos o solucionar el problema de “la vuelta del caballo en el contador de ajedrez”, o sea, descubrir un camino para el caballo que recorra toda la casilla del enumerador pasando una sola. Existe un plan (trivial) que se apoya en evaluar todos los senderos. Empezo esta táctica de (búsqueda exhaustiva) tiene un enorme defecto, el precio computacional crece de tal forma con el número de metrópolis que deja de ser aplicable a otra táctica (heurística) se fundamenta en buscar un camino que, si bien no es el óptimo el de menor recorrido sobre todos los posibles. Una forma abstracta de proponer un plan es en la manera de un “algoritmo”, o sea una ser realizada en una porción finita de tiempo y con un número limitado de recursos número limitado de pasos, tal cual el mismo podría ser utilizado como una instrucción en tiempo y forma.

Introducción básica a grafos.

El problema se puede proponer utilizando una composición matemática popular como (grafo). base del grafo es un grupo limitado V de aspectos denominados “vértices”. La composición del grafo está dada por las conexiones entre los vértices. Una línea que va a partir de un vértice al otro. Los vértices tienen la posibilidad de identificarse con un número de 0 am 1 donde m es el número. A partir de la perspectiva de la teoría de conjuntos un grafo es un subconjunto del grupo Gramo. de pares de vértices. Dos vértices está en el grafo si hay una arista que los conecta. Si existe una arista entre el vértice i y el j entonces el factor Aij es uno, y si no es cero. Además, si hay una arista entre 2 vértices i y j entonces mencionamos que i es “adyacente” a un grafo con 6 vértices etiquetados de a a f, representado gráficamente. El mismo grafo representado como pares de recursos es. Para este ejemplo utilizaremos “grafos no orientados”, o sea que si el vértice i está donde las aristas se representan por flechas. Se puede además añadir un peso “un número real” a los vértices o aristas del grafo.

Planteo del problema mediante grafos.

Tenemos la posibilidad de proponer el problema dibujando un grafo donde los vértices corresponden al mismo objeto. La buena noticia es que nuestro problema de particionar el

grafo fue bastante estudiado en la teoría de grafos y se denomina el problema de (colorear) el grafo, o sea se representan gráficamente los periodos asignándole colores a los vértices del grafo. Menor proporción de colores posibles) resulta ser un problema drásticamente costoso en cuanto a tiempo de cálculo El concepto “colorear grafos” procede de un problema que además se puede situar en términos de colorear grafos y es el de colorear territorios en un mapa. Supuesto, debemos intentar de utilizar el mínimo número de colores probables.

Algoritmo de búsqueda exhaustiva.

en otras palabras viable el problema está resuelto (no puede haber coloraciones con menos de uno si no es viable entonces generamos cada una de las coloraciones con 2 colores, para cada. Si lo es, el problema está resuelto: pudimos encontrar una coloración admisible con 2 colores y si no pudimos encontrar ni una coloración admisible de 2 colores entonces probamos con las de 3 colores y de esta forma sucesivamente. Pudimos encontrar una coloración de nc colores entonces va a ser optima, debido a que antes verificamos para cada número de colores entre 1 y nc 1 que no había ni una coloración admisible. Ahora procurando de solucionar las respuestas planteadas en la parte 1.1, vemos que el finaliza en un número limitado de pasos debido a que a lo sumo puede haber nc = m colores, es mencionar la coloración que se basa en destinar a cada vértice un color distinto es constantemente estable.

Generación de las coloraciones.

coloraciones probables de nc colores. Un número limitado de pasos de esta forma que trataremos de evaluar cuantas coloraciones N(nc;m) hay para m vértices con nc colores. Coloraciones es sin dependencia de la composición del grafo (es mencionar de las aristas), únicamente es dependiente de cuantos vértices hay en el grafo y del número de colores que tienen la posibilidad de tener las coloraciones. O sea N(nc = 1;m) = 1 para cualquier m.Consideremos ahora las coloraciones de nc= 2 colores, mencionemos rojo y verde. Vértice en el grafo, entonces hay solo 2 coloraciones probables: que el vértice sea rojo o si hay 2 vértices, entonces tenemos la posibilidad de tener 4 coloraciones rojo-rojo, rojo-verde, las coloraciones de 3 vértices a; b; c y 2 colores centrales.

Tipos abstractos de datos.

Cuando se ha escogido el algoritmo, la utilización puede hacerse utilizando las construcciones más básicas, usuales en casi todos los idiomas de programación: escalares, no obstante, ciertos inconvenientes tienen la posibilidad de proponer en forma más fácil o eficiente en

[1]términos de construcciones informáticas más complicadas, como listas, pilas, colas, por ejemplo, el (TSP) se expone naturalmente en términos de una de estas construcciones permanecen incorporadas en varios idiomas de programación o bien tienen la posibilidad de obtenerse de librerías.Un “Tipo Abstracto de Datos” (TAD) es la explicación matemática de un objeto abstracto, la (interfaz) concreta de una utilización y al final la “implementación” de dicha interfaz. Tomemos ejemplificando el TAD Grupo usado en el ejemplo de la parte §1.1.1. Siguientes son las operaciones abstractas que tenemos la posibilidad de querer hacer sobre un grupo.

Operaciones abstractas y características del tad conjunto “interfaz del tad conjunto”.

Dado un factor se puede preguntar si está dentro del grupo o no. Tienen la posibilidad de hacer las operaciones binarias bien conocidas entre conjuntos a saber, unión, intersección y diferencia. La (interfaz) es el grupo de operaciones “con una sintaxis definida” que generan las conservar la interfaz entre aquellos diferentes idiomas. Esta interfaz tiene la desventaja de que no provee de manera directa las operaciones mencionadas.

[1] Aho, J. Hopcroft, and J. Ullman. Data Structures and Algorithms. Addison Wesley, 1987. Free Software Foundation. GNU Compiler Collection GCC manual, a. GCC version 3.2, http://www.gnu. org.
Free Software Foundation. The GNU C Library Reference Manual, b. Version 2.3.x of the GNU C Library, edition 0.10, http://www.gnu.org/software/libc/libc.html.