



UNIVERSIDAD NACIONAL ABIERTA  
VICERRECTORADO ACADÉMICO  
ÁREA INGENIERÍA

TRABAJO PRÁCTICO:

ASIGNATURA: **COMPUTACIÓN II**

CÓDIGO: **324**

FECHA DE ENTREGA DE LAS ESPECIFICACIONES AL ESTUDIANTE: **A partir de la primera semana de aplicación de pruebas, a través del asesor de la asignatura en su centro local**

FECHA DE DEVOLUCIÓN DEL INFORME POR EL ESTUDIANTE:  
**Adjunto a la Prueba Integral**

NOMBRE DEL ESTUDIANTE:

CÉDULA DE IDENTIDAD:

CORREO ELECTRÓNICO:

TELÉFONO:

CENTRO LOCAL:

CARRERA: **236**

NÚMERO DE ORIGINALES:

FIRMA DEL ESTUDIANTE:

LAPSO: **2015-2**

**UTILICE ESTA MISMA PÁGINA COMO CARÁTULA DE SU TRABAJO**

**RESULTADOS DE CORRECCIÓN:**

OBJ N°		5	6	7
0:NL	1:L			

## ESPECIFICACIONES DEL TRABAJO PRÁCTICO

Especificaciones: Este trabajo se basará en el Objetivo No. 5 del Módulo II y los objetivos 6 y 7 correspondientes al Módulo III. En él se evidenciará las habilidades y destrezas adquiridas por el estudiante, al momento de implementar algoritmos para la resolución de problemas empleando la estructura de Grafos, así como algoritmos de ordenación y búsqueda en Lenguajes de Programación.

### Objetivo 5

- 1- El problema de los cuatro colores parte de lo siguiente: bastan cuatro colores para colorear un mapa geopolítico plano, sin que dos países con frontera común tengan el mismo color. Un mapa es siempre conexo y cada una de sus regiones también lo es. Dos regiones no pueden tocarse en un único punto. Este problema es uno de los más famosos en matemáticas, y tiene la característica de haber interesado no sólo a matemáticos puros sino también a computistas.

El problema fue propuesto en 1852 por un estudiante, Francis Guthrie, quien plantea el problema a su hermano Frederick y a Augustus De Morgan, cuando observó que podía colorear el mapa de los condados ingleses utilizando sólo cuatro colores. En la coloración cada condado recibía un color y dos condados que compartían frontera (no sólo un punto) recibían colores diferentes. Durante más de cien años los intentos de prueba de la Conjetura de los cuatro colores generaron importantes avances en la investigación en grafos, hasta que en 1976 Appel y Haken anunciaron que habían resuelto el problema con la ayuda del computador, empleando cincuenta días de cálculo. El enunciado del problema es el siguiente: “¿Cuántos colores son necesarios para dibujar un mapa político, con la condición obvia que dos países adyacentes no puedan tener el mismo color? Se supone que el mundo es esférico o plano. La respuesta a este teorema es que con solo cuatro colores son siempre suficientes para colorear cualquier mapa. La forma de cada país o estado no importa; lo único importante es saber qué país toca a qué otro.

El mapa expuesto en la Figura N° 1 muestra que tres colores no bastan para cumplir con la condición de no repetición de colores en estados contiguos. Los colores se han etiquetado con los números del uno al cuatro.

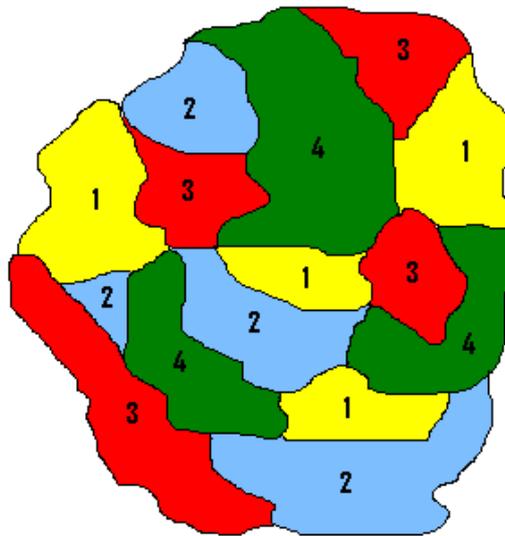


Figura 1

Color 1: amarillo, 2: azul, 3: rojo, 4: verde

Un **grafo no dirigido** es un conjunto de objetos denominados vértices unidos por enlaces denominados arcos o aristas, que se emplean para representar relaciones binarias. En términos matemáticos se representa como  $G = (V, A)$ , en donde  $V$  es el conjunto de vértices y  $A$  el conjunto de arcos o aristas. Hay muchos problemas, como la asignación de tareas, los problemas de almacenamiento y la coloración de mapas entre otros que se pueden modelar empleando la teoría de grafos. Los elementos son: un conjunto de vértices y otro de aristas de tal forma que los vértices (aristas) adyacentes pertenezcan a diferentes conjuntos de la partición. Tales particiones se llaman coloraciones (coloraciones de aristas). Los problemas sobre coloración de grafos fueron, en la segunda mitad del siglo XIX, uno de los hitos iniciales de la Teoría de Grafos.

Existen varios tipos de coloraciones que se pueden realizar sobre los grafos: Coloraciones de vértices, de aristas, de caras.

En términos generales, una coloración de un grafo  $G(V, A)$  por vértices, es una asignación de colores a los vértices de  $G$ ; es decir, a cada vértice un color de forma que vértices adyacentes reciban colores distintos. Si en la coloración se usan  $k$  colores se dice que es una  $k$ -coloración. Las coloraciones siempre existen, pues podemos asignar a cada vértice del grafo un color diferente si fuese necesario. Cada coloración de  $G$  produce en  $V(G)$  una partición en conjuntos independientes denominados clases de color. Si existe una  $k$ -coloración de  $G$  se dice que el grafo  $G$  es  $k$ -coloreable. El mínimo  $k$  para el que un grafo  $G$  es  $k$ -coloreable se llama número cromático de  $G$ , y se designa por  $\chi(G)$ .

Afirmar que  $\chi$  es el número cromático de un grafo significa que podemos colorear el grafo con  $\chi$  colores y que no existe una coloración con menor número de colores. Si un grafo  $G$  tiene  $n$  vértices siempre podemos colorear los vértices de  $G$  con  $n$  colores, uno distinto para cada vértice, por lo que  $\chi(G) \leq n$ .

### **Métodos de coloración de grafos**

No es fácil determinar el número cromático de un grafo. Sin embargo, existen algoritmos eficientes para colorear grafos de forma que el número de colores usados esté "próximo" a su número cromático. Las heurísticas que se utilizan en estos algoritmos son las siguientes:

- Colorear un vértice de grado alto es más difícil que otro de grado bajo.
- Los vértices con los mismos vecinos deben colorearse al mismo tiempo.
- Asignar a muchos vértices el mismo color es una buena idea.

El grado (o valencia) de un vértice  $v$  se define como el número de aristas que inciden en él, y se denota con  $g(v)$ . Si  $g(v)=0$ , se dice que  $v$  es un vértice aislado.

A continuación se esboza el algoritmo de coloración de Welsh - Powell

### Algoritmo Welsh - Powell

---

**Paso 1:** Hallar la valencia de cada vértice.

**Paso 2:** Listar los vértices en orden decreciente de valencia.  
Si hay empates aplique el criterio que desee.

**Paso 3:**  $i \leftarrow 1$ .

**Paso 4:** Colorear el **primer vértice** de la lista con el color  $i$ .

**Paso 5:** Continúe escogiendo todos los vértices de la lista **no conectados** con el vértice en cuestión y asignándoles el mismo color.

**Paso 6:** Tache todos los vértices coloreados de la lista.

**Paso 7:**  $i \leftarrow i + 1$ .

**Paso 8:** Si aún quedan vértices sin colorear, vaya al Paso 4.  
Si no, **Fin**

---

Ejemplo: En la figura N° 2 se representa un grafo.

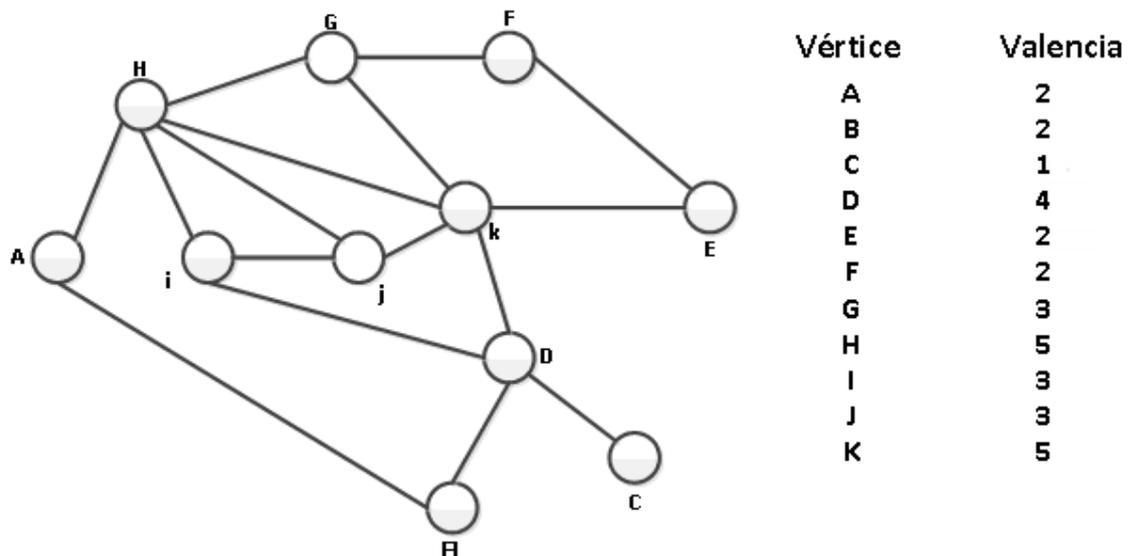


Figura 2

Al ordenar los vértices de acuerdo a su valencia se tiene la siguiente lista L: H, K, D, G, I, J, A, B, E, F, C.

Al aplicar el algoritmo obtendremos en la primera iteración, lo siguiente:

- Colorear con rojo (color 1) los nodos: H, D y E. Eliminarlos de la lista, quedando L: K, G, I, J, A, B, F, C.
- Colorear con azul(color 2) los nodos: K, I, A, F, C. Resultando L: G, J, B.
- Colorear con amarillo (color 3) los nodos: G, J, B.

El grafo coloreado se muestra en la figura N° 3, en donde los colores se especifican dentro de los nodos. En este caso sólo se necesitaron 3 colores, luego 3 es el número cromático.

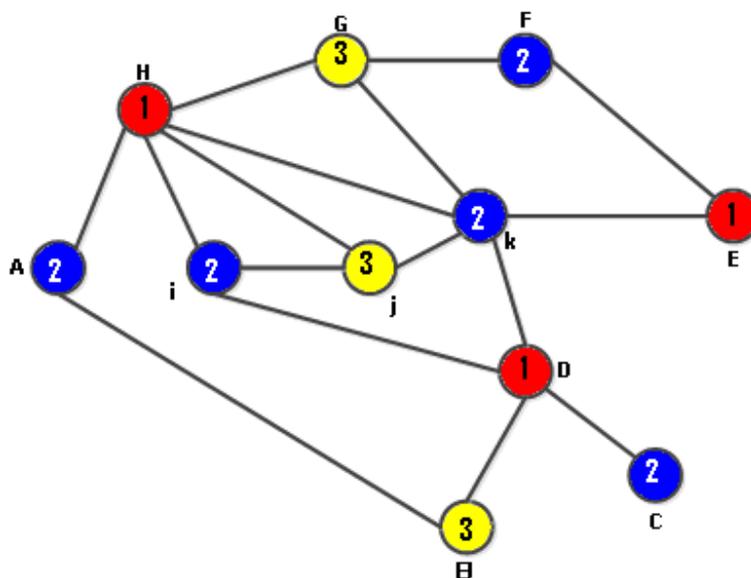


Figura 3

Sobre la base de la información dada sobre coloración de grafos, se pide realizar un programa en lenguaje de programación C++( o Dev C++) que permita colorear un grafo de N nodos ( $20 \geq N \geq 10$ ):

- El programa solicitará el número de vértices y generará las aristas de manera aleatoria. El procedimiento de generación de la matriz de Adyacencia se especificará más abajo.

- Aplicará el algoritmo de Welsh - Powell para colorear los vértices o nodos.
- Imprimirá inicialmente: el valor de N y la matriz de adyacencia y al final los nombres de los colores asignados a cada nodo.

### Procedimiento para la generación de la Matriz de Adyacencia

La matriz de Adyacencia debe crearse automáticamente, una vez solicitado el valor de N, utilizando un proceso de generación de números aleatorios (NA) para determinar si entre cada par de nodos  $i, j$  existe o no un arco. Un NA es un valor de una variable obtenido al azar, dicha variable se especifica según una función (denominada función de distribución).

En el lenguaje de programación se puede emplear la función  $rand()$ , para generar números aleatorios entre 0 y 1. Se sugiere implementar ésta, como se describe a continuación:

$$f(\text{NA}) = \begin{cases} 0 & \text{si } 0 \leq \text{NA} \leq 0,5 \\ 1 & \text{si } 0,5 < \text{NA} \leq 1 \end{cases}$$

De esta manera si se quiere determinar la relación existente entre los nodos  $i$  y  $j$ , y obtenemos un valor de  $f = 0$  se deduce que no hay un arco entre  $i$  y  $j$  y si el valor de  $f$  es 1, se deduce que hay un arco entre  $i$  hasta  $j$ . Empleando este proceso para cada par de nodos, se construye la Matriz de Adyacencia. Recuerde que esta matriz es simétrica, por lo cual debe emplear una función eficiente para su construcción.

### Criterio de corrección

Se considera logrado el objetivo si al menos se cumple con lo siguiente:

- ✓ Entrega del listado documentado del programa, codificado en C++ en forma modular y estructurada. En el encabezado de cada función o sección de programa que lo requiera y en la declaración de las estructuras de datos se incluye un breve comentario acerca del proceso, método o definición de estructura, según sea el caso.

- ✓ El programa **corre** sin restricciones. En general cada programa incluye funciones que realizan o contribuyen a alcanzar lo solicitado en las especificaciones.
- ✓ Imprime los resultados exigidos.

## 2- OBJETIVO 5

Elabore un programa en C++ que realice lo siguiente:

- Genere de manera aleatoria, números que tengan hasta 3 dígitos. Utilice la función *rand()*.
- Ordene esta colección de datos, empleando los métodos: **Quicksort** y **Heapsort**, almacenando los datos en una estructura de datos apropiada.
- Imprima los resultados antes y después de ordenarlo.

### Criterio de corrección

Se considera logrado el objetivo si al menos se cumple con lo siguiente:

- ✓ Entrega del listado documentado del programa, codificado en C++, en forma modular y estructurada. En el encabezado de cada función o sección de programa que lo requiera y en la declaración de las estructuras de datos se incluye un breve comentario acerca del proceso, método o definición de estructura, según sea el caso.
- ✓ El programa corre sin restricciones. En general cada programa incluye módulos que realizan o contribuyen a alcanzar lo solicitado en las especificaciones.
- ✓ Imprime los valores a ordenar, obtenidos de manera aleatoria y al final los valores ordenados al aplicar cada método.

## 3- OBJETIVO 6

Elabore una función en C++ de **búsqueda**, tal que dado un proceso continuo de lectura de datos, los busque en la estructura ordenada, obtenida por alguno de los métodos implementados en la sección anterior y determine si se encuentra en ella; en caso contrario emita el mensaje apropiado. Si el dato leído está en la estructura, imprimirá el elemento anterior y el posterior presente en la estructura ordenada, si acaso existen.

### **Criterio de corrección**

Se considera logrado el objetivo si al menos se cumple con lo siguiente:

- ✓ Entrega del listado documentado del programa, codificado en C++, en forma modular y estructurada. En el encabezado de cada función o sección de programa que lo requiera y en la declaración de las estructuras de datos se incluye un breve comentario acerca del proceso, método o definición de estructura, según sea el caso.
- ✓ El programa **corre** sin restricciones. En general cada programa incluye módulos que realizan o contribuyen a alcanzar lo solicitado en las especificaciones.
- ✓ Imprime los resultados exigidos.

### **Instrucciones generales sobre el Trabajo Práctico**

El estudiante debe entregar lo siguiente:

- Listado documentado del programa fuente. En el encabezado de cada función o sección de programa que lo requiera, debe incluir un breve comentario del proceso que se realiza o del método que aplica. Igualmente es conveniente hacerlo en la definición de las estructuras de datos y variables utilizadas.
- Listado de los resultados.
- CD (Disco Compacto) que contenga el programa fuente (.CPP) y el programa ejecutable (.EXE), debidamente identificado.

### **Recomendaciones**

- Emplee nombres de variables, constantes y funciones alusivos a lo que representan.
- Utilice un diseño modular para la resolución del problema. Esta estructura aportará legibilidad y facilidad de comprensión, además evitará redundancias en los procesos. Evite variables globales en las funciones. Emplee parámetros en los mismos, determine cuáles son *parámetros valor* y cuáles *parámetros variables*.
- Desarrolle algoritmos eficientes.
- Elabore funciones de **validación** de la data y de detección de errores para evitar interrupciones inesperadas en la ejecución del trabajo.
- El CD debe estar libre de virus y debe entregarse en un sobre conjuntamente con el listado de programa y resultados. No use cinta engomada para adherir el CD.
- El trabajo se entregará completo, adjunto a la prueba integral, con una portada similar a la presentada en las especificaciones de este trabajo.

#### FIN DE LAS ESPECIFICACIONES DEL TP

NOTA: Los Trabajos Prácticos son estrictamente individuales y una producción inédita del estudiante, cualquier indicio que ponga en duda su originalidad, será motivo para su anulación. Queda a discreción del asesor o profesor corrector, solicitar una verificación de los objetivos contemplados en el mismo, únicamente en aquellos casos en los que se vea comprometida la originalidad de la autoría del presente trabajo práctico.