

MODELO DE PREGUNTAS

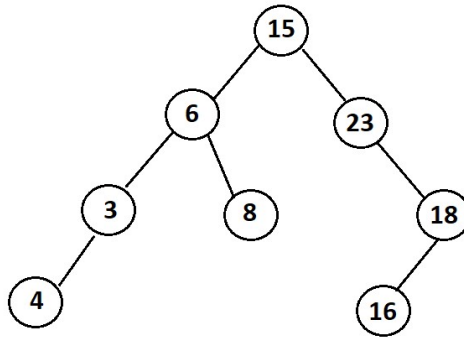
Nota: lo que está entre corchetes significa que puede ir una de las opciones que se incluyen.

LISTAS

- 1) Implementar en C++ o pseudocódigo el método [insertar / eliminar] un elemento x de una lista [SE / DE / circular-SE / circular-DE] en cierta posición.
- 2) Implementar en C++ o pseudocódigo el método [insertar / eliminar] un elemento x de una lista [SE / DE / circular-SE / circular-DE], dado que la lista está ordenada por la clave de los elementos ($x.clave$).
- 3) Implementar en C++ o pseudocódigo el método [insertar / eliminar] un elemento x de una [Pila / Cola].

ÁRBOLES

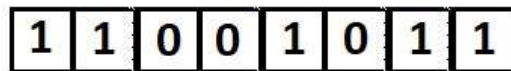
- 4) Indicar cómo queda un ABB en cada estado, sabiendo que comienza vacío y se hacen las siguientes operaciones:
 - a. a.insertar(15)
 - b. a.insertar(8)
 - c. a.insertar(21)
 - d. a.insertar(18)
 - e. a.insertar(3)
 - f. a.eliminar(21)
 - g. a.insertar(12)
 - h. a.insertar(16)
 - i. a.insertar(10)
 - j. a.eliminar(8)
- 5) Explicar qué es un AVL y cómo funciona. Cuáles son sus ventajas y desventajas.
- 6) Definir árbol B de orden m .
- 7) Sea un árbol B de orden $m = 5$. Indicar cómo queda en cada paso al insertarse las siguientes claves: 6, 11, 5, 4, 8, 9, 12, 21, 14, 10, 19, 28, 3, 17, 32, 15, 16, 26 y 27.
- 8) Dado el árbol:



a. Indicar si es un ABB. Justificar.

b. Indicar cómo queda la salida con un recorrido en [pre orden / post orden / in orden / ancho primero]

9) Sea el array de bits



a. Indicar qué elementos se encuentran.

b. Qué pasos y operaciones se deben hacer para

i. Dar de alta al valor 4.

ii. Dar de baja al valor 3.

iii. Consultar si está el valor 5.

10) Similar anterior pero con un array de 2 enteros (asumiendo 4 bytes por c/u).

11) Explicar qué es un heap de mínima y qué es un heap de máxima.

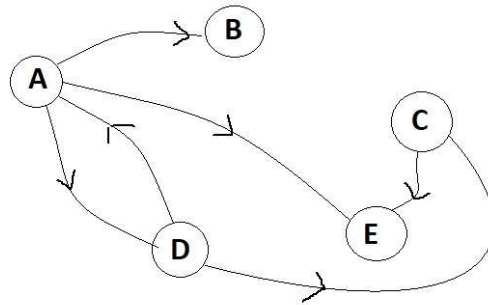
12) Se implementa un heap de mínima en un vector

a. Si llegan los siguientes valores: 15, 3, 18, 21, 32, 4, 9, 1, indicar cómo va quedando el vector (graficar también el árbol heap en cada inserción)

b. Si se quita el mínimo ¿cómo se reestructura el heap? Indicar cada paso con un gráfico del vector y del árbol.

GRAFOS

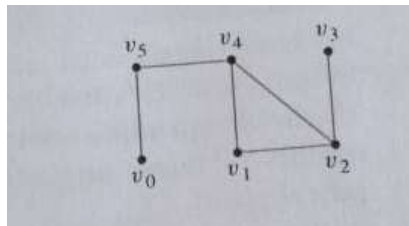
13) Dado un grafo dirigido:



- a. Indicar cómo es la matriz de adyacencia.
- b. Indicar cómo es la lista de adyacencia.
- c. Indicar cómo queda el recorrido en profundidad primero. Imprimir los vértices y explicar el recorrido habiendo elegido la matriz de adyacencia o la lista de adyacencia.
- d. Indicar cómo queda el recorrido en ancho primero. Imprimir los vértices y explicar el recorrido habiendo elegido la matriz de adyacencia o la lista de adyacencia.

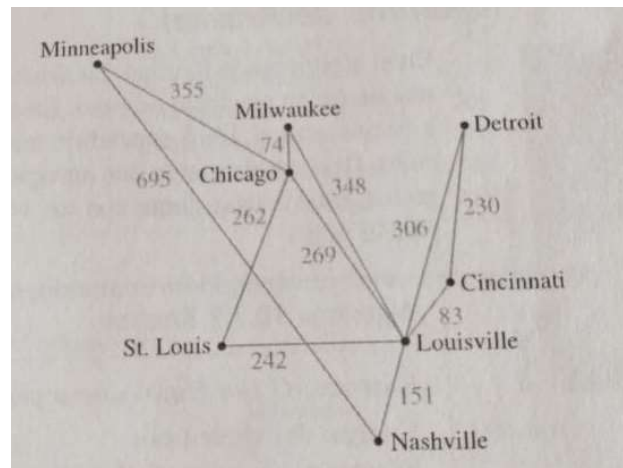
14) Definir árbol de expansión mínimo de un grafo.

15) Dado el siguiente grafo

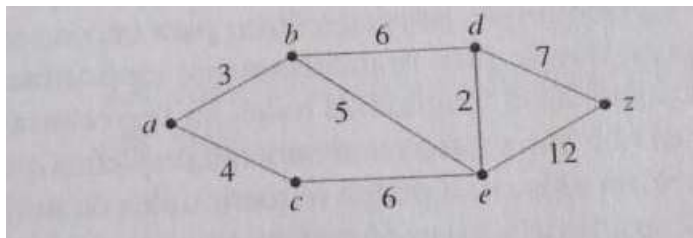


Encontrar todos los árboles de expansión mínimo.

16) Dado el siguiente grafo, donde los vértices indican ciudades y los números de los arcos son las distancias, encontrar el árbol de expansión mínimo, usando el algoritmo de [Kruskal / Prim](#).



17) Dado el grafo



Hallar el camino menos costoso para ir desde a hasta z usando el algoritmo de Dijkstra.

ORDENAMIENTO

18) Sea el vector $v = [150, 31, 83, 325, 46, 186, 222]$

- Explicar cómo funciona el algoritmo de ordenamiento [shell / radix / quick / merge / heap / tree].
- Ordenar el vector, indicando cómo queda en cada paso, aplicando dicho algoritmo.

COMPLEJIDAD

19) Dado el algoritmo:

```
int i, k, a;
for (a = 0; a < 10; a++) {
    i = rand () % 10;
    if (i >= 5)
        k = i + 30;
    else
        k = 2;
```

```

        cout << i << " " << k << endl ;
    }

```

- a. Indicar las OE para el mejor caso.
- b. Indicar las OE para el peor caso.
- c. Indicar su orden O para el peor caso.

20) Dado el algoritmo recursivo: $T(n) = 2T\left(\frac{n}{2}\right) + 3$

- a. Encontrar la solución del costo en forma no recursiva por el método de expansión.
- b. Encontrar la solución del costo utilizando el Teorema Maestro (se puede llevar la copia).

RECURSIVIDAD

- 21) Escribir en C++ o pseudocódigo un algoritmo recursivo para sumar los elementos de un vector de enteros (el vector y su tamaño se pasan por parámetro).
- 22) Escribir en C++ o pseudocódigo un algoritmo recursivo aplicando DyV para sumar los elementos de un vector de enteros (el vector y su tamaño se pasan por parámetro).

HASHING

- 23) ¿Qué es hashing? ¿Para qué se usa?
- 24) Nombre y explique brevemente las funciones de hashing que conoce.
- 25) ¿Qué es una colisión? ¿Qué formas de resolverla conocés? Nombrar y explicar.
- 26) Si se usa módulo de 11 en una tabla de tamaño 11 y la solución de colisiones que prefieras, indicar cómo funciona cada operación y cómo va quedando la tabla en cada paso: insertar(37), insertar(12), insertar(8), insertar(26), eliminar(37), insertar(34), buscar(26)