Algoritmos y Estructuras de Datos

Examen Final
Pregrado
2022-2

Indicaciones específicas:

Duración: 90 minutos + 10 minutos para subir sus respuestas
Número de preguntas: 3

- **No se permitirá consultar paginas externas al Canvas**
- **No se permitirá compartir notas ni material entre alumnos.**
- Lea las preguntas cuidadosamente y responda de manera clara. Respuestas que no sean legibles o claras no tendrán ningún puntaje.
- Si no sabe la respuesta, deja el espacio en blanco y coloque **"F"**.
- Recuerden, de todas formas, resolver las preguntas que no entendieron después del examen.

Pregunta 1 (4 puntos): Preguntas

A. (2 pts) ¿Qué estructuras de datos (de las vistas en el curso) me permiten aplicar la **búsqueda por rango** de manera eficiente?

Estructura de Datos	¿Por qué?	Complejidad

B. (1 pts) Sparse Matrix puede implementarse de manera eficiente usando Hashing, ¿Cuál sería su ventaja respecto a su versión con Linked Lists?.

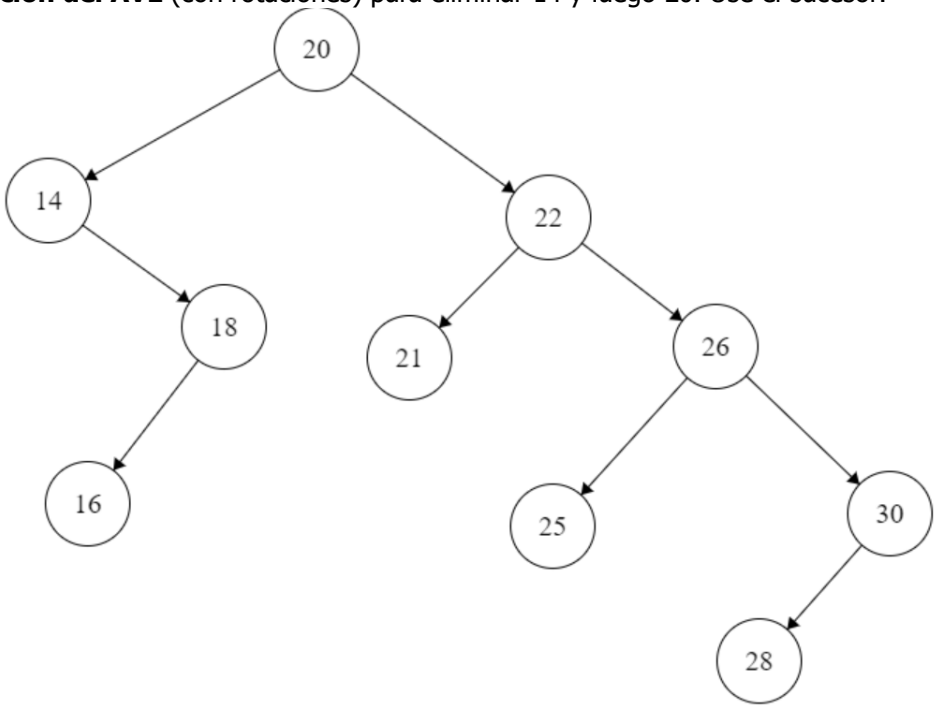
C. (1 pts) Al aplicar el método del Backtraking para resolver el problema de la Mochila ¿Cuáles serias las cotas inferior y superior del algoritmo?

CI: _

CS: _

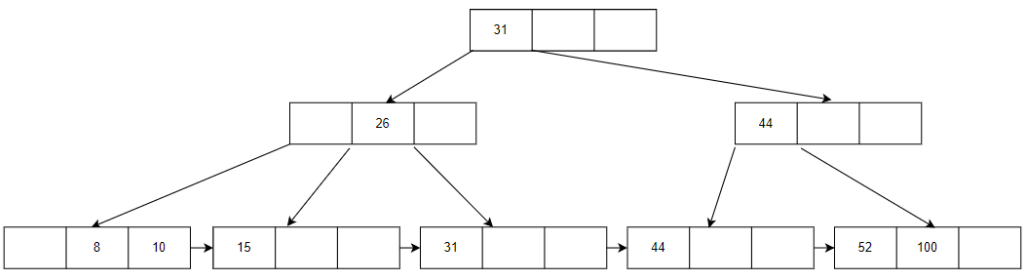
Pregunta 2 (8 puntos): Ejercicios

A. (1.5 pts) AVL: dado el siguiente BST no balanceado, se le pide aplicar el algoritmo de **eliminación del AVL** (con rotaciones) para eliminar 14 y luego 20. Use el sucesor.



1- Después de eliminar 14	2- Después de eliminar 22

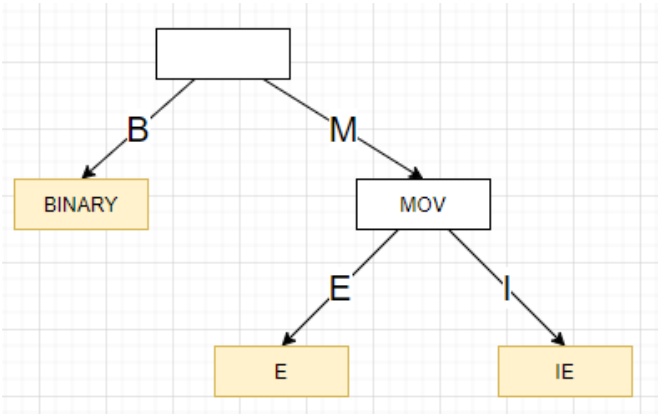
B. (2.5 pts) B+: complete el árbol para que se cumpla las propiedades del B+ y luego eliminar el 31 y 44



- El criterio de distribución en el Split para una cantidad impar, seria asignar más elementos al nodo izquierdo.
- Una llave en un nodo interno, es el máximo del subárbol izquierdo.

1- Completar elementos del árbol
2- Eliminar 31 y 44

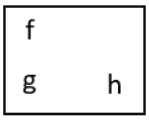
C. (1.5 pts) Dado el siguiente árbol Patricia, se le pide insertar BIN y MOVEMENT. Dibujar después de cada inserción. El nodo amarillo indica fin de una palabra.



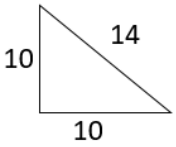
D. (2.5 pts) Aplique el algoritmo A* en el siguiente laberinto.

- Inicio: celda en verde
- Objetivo: celda en rojo
- Los cuadros grises son muros y no deben ser considerados en la solución.

Etiquetado de valores

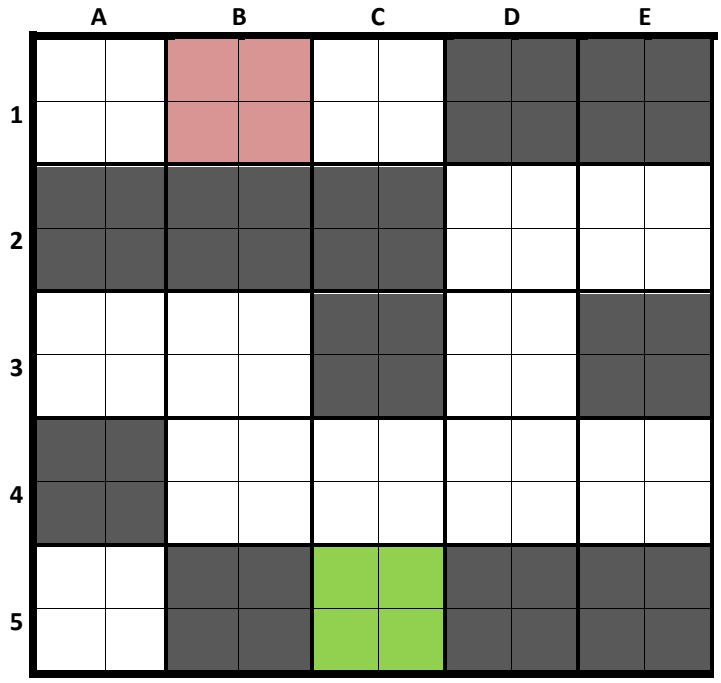


Costo de movimiento



Heurística
Distancia Manhattan

- **Anote dentro de la tabla todos los valores f(n), g(n) y h(n), y también los ancestros.**
- **Indicar el camino final**



Pregunta 2 (8 puntos): Implementación

A. (4 puntos) Patricia Tree: se le pide diseñar un seudocódigo lo más detallado posible para transformar un Trie Simple a un Trie Compactado. La idea es cubrir todos los casos de eliminación y analizar la complejidad algorítmica en el código.
N: total de elementos en el Trie.
E: tamaño del alfabeto

B. (4 puntos) Sparse Matrix: se le pide diseñar el algoritmo en C++ para eliminar una columna de una matriz lo más eficiente posible. Considerar todos los casos.

```
struct SparseMatrix {
    struct Node {
        T data;
        int pos_row;
        int pos_col;
        Node<T>* next_row;
        Node<T>* next_col;
    };
    vector<Node<T>*> rows;
    vector<Node<T>*> cols;
    int n_cols;
    int n_rows;
};
//eliminar la columna col de la matriz original
void remove_column(SparseMatrix &ma, int col);
```