Algoritmos y Estructuras de Datos

Examen Final
Pregrado
2023-0

Indicaciones específicas:

Duración: 90 minutos + 10 minutos para subir sus respuestas
Número de preguntas: 5

- Se permite el uso de calculadoras, y hojas en blanco de ayuda.
- **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) Un diccionario (colección de elementos **key:value**) puede gestionarse con diferentes estructuras de datos. Realice un breve comparativo diferencial entre tres de ellas.

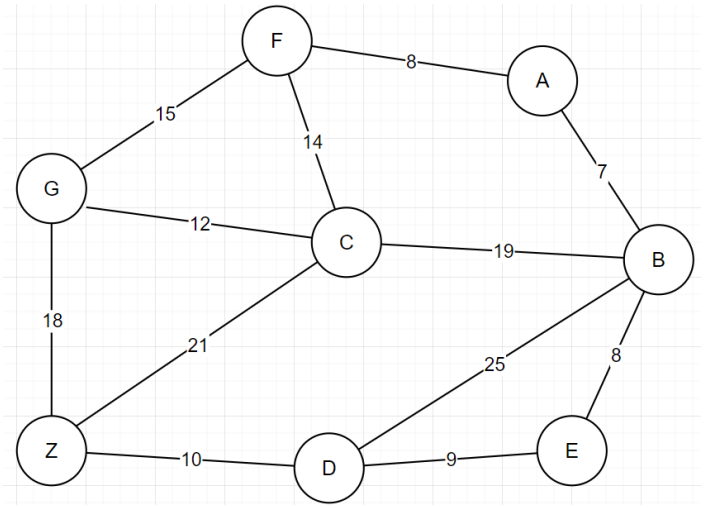
Estructura de datos	**Características diferenciales**

B. (2 pt) ¿Qué estructuras de datos (de las que estudiaron en el curso) me permiten aplicar la **búsqueda por rango** de manera eficiente?

Estructura de Datos	¿Cómo lo hace?	Complejidad

Pregunta 2 (11 puntos): Ejercicios

A. (2.5 puntos) Dado el siguiente grafo ponderado, aplicar el algoritmo de Kruskal e indique de forma ordenada las aristas que se van agregando a la solución ({A, D}, {C, F}, …) y el DisjoinSet resultante (optimización by rank).

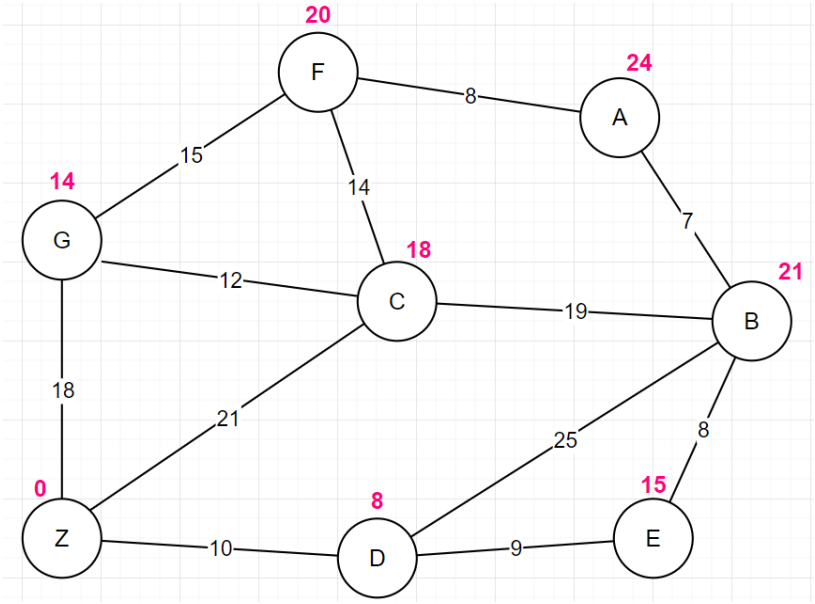


Orden de aristas Kruskal:

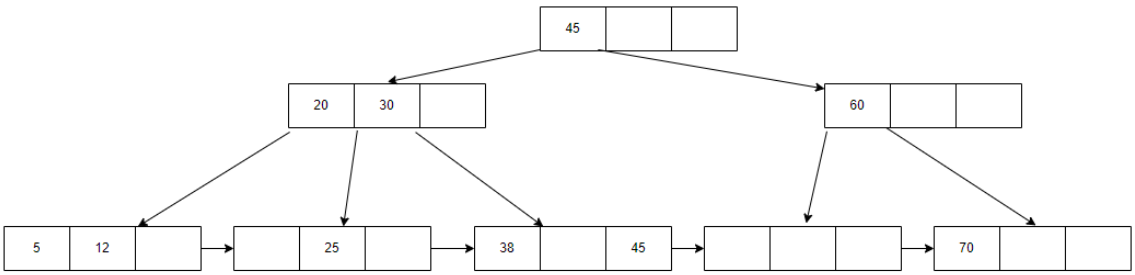
DS cuando se inserta la 3ra arista:

DS cuando se inserta la 6ta arista:

B. (2.5 puntos) Dado el siguiente grafo ponderado, se le pide encontrar el camino más eficiente para llegar **desde 'A' hasta 'Z'** utilizando el algoritmo de **A*.** Los números en las aristas representan la distancia entre vértices. Los números encima de los nodos representan el valor de la heurística. Utilice cualquier método de resolución paso a paso.



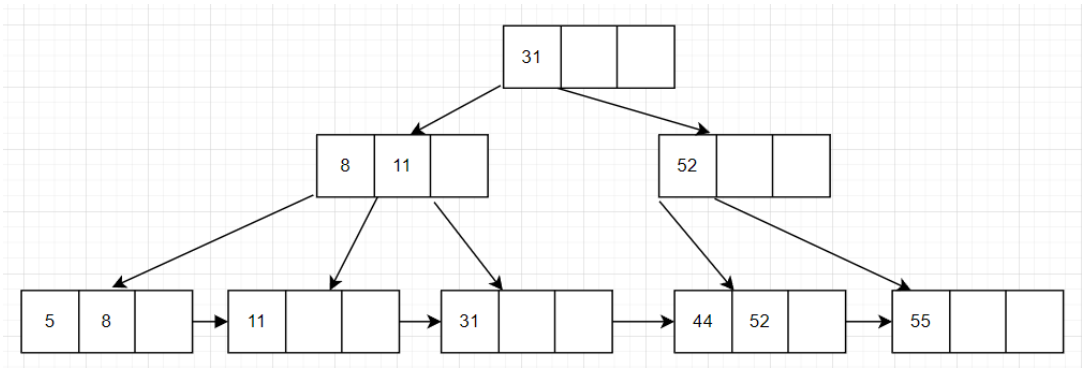
C. (2 puntos) complete el árbol (sin alterar los elementos existentes) para que se cumpla las propiedades del B+ Tree y luego insertar 15 y 35



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

1- Completar elementos del árbol
2- Insertar 15 y 35

D. (2 puntos) Dado el siguiente B+ Tree elimine los siguientes elementos. Dibuje después de cada Eliminación.
Eliminar: 11,31,5

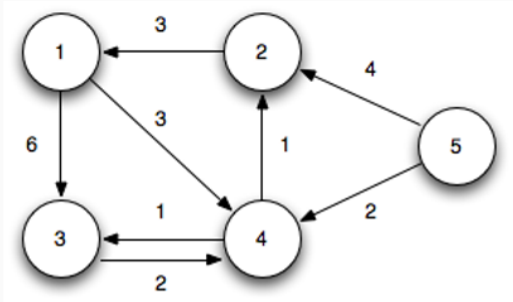


- Una llave en un nodo interno, es el máximo del subárbol izquierdo.

E. (2 pts) Suffix Tree: dado el texto "anahabana", construir el árbol de sufijos compactado y luego señalar el camino del string matching para "ana".

Pregunta 3 (7 puntos): Implementación

A. (3 pts) Diseñe el algoritmo en C++ para construir el camino a partir de la matriz de recorridos resultante del Floyd Warshall.



Matriz de Recorridos Final

	1	2	3	4	5
1	-	4	4	4	5
2	1	-	4	1	5
3	4	4	-	4	5
4	2	2	3	-	5
5	4	4	4	4	-

```
List<int> get_path(int tour_matrix[N][N], int begin, int end);
```

A. (4 pts) Sparse Matrix: se le pide diseñar el algoritmo en C++ para devolver la suma de dos matrices lo más eficiente posible.

```
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;
};

//devolver una nueva matriz como la suma de ambas
SparseMatrix transpose(SparseMatrix matriz1, SparseMatrix matriz2);
```