Resolución de Tarea 5 - Grafos (Fecha: 3 de Noviembre de 2025)

Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI5651 - Diseño de Algoritmos I
Septiembre - Diciembre 2025
Estudiante: Junior Miguel Lara Torres (17-10303)

Tarea 5 (9 puntos)

Indice

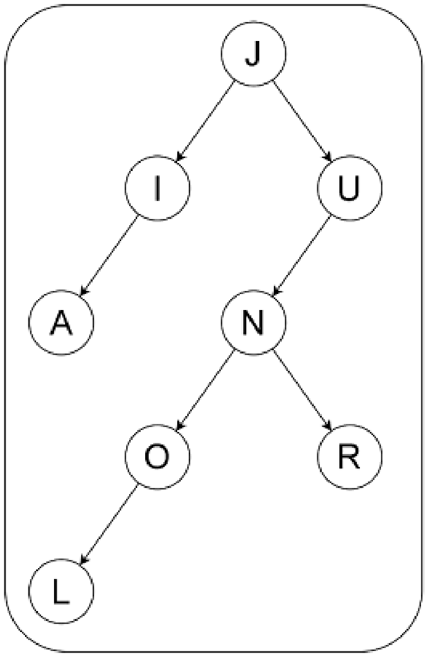
Pregunta 1

Mi nombre y apellido es: **Junior Lara**

- Cadena completa en minúsculas: "juniorlara".
- Eliminación de repeticiones (conservando el orden de aparición): **j, u, n, i, o, r, l, a**
- Cadena de caracteres resultante (S): "juniorla" (n=8 caracteres).

Parte (a)

El árbol binario de búsqueda se muestra a continuación

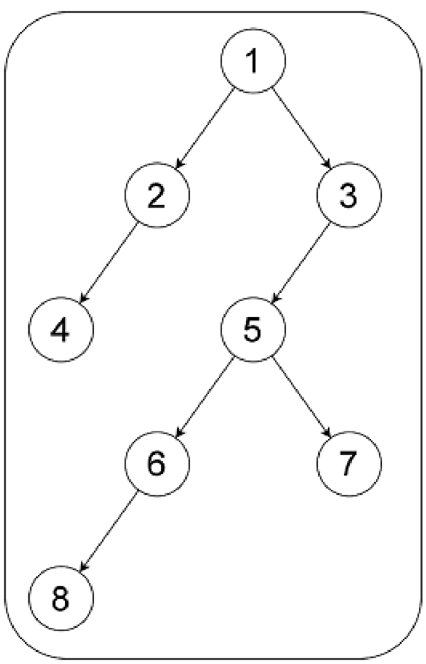


Parte (b)

El recorrido preorder es el siguiente: **j, i, a, u, n, o, l, r**

Parte (c)

Realizando la enumeración por niveles



- El recorrido de Euler es el siguiente: **1, 2, 3, 2, 1, 2, 3, 4, 5, 5, 3, 4, 3, 2, 1**

- A nivel de caracteres tenemos: **j, i, a, i, j, u, n, o, l, o, n, r, n, u, j**

Parte (d)

La cadena es "juniorla". Los nodos a considerar son 'l' y 'a'.

El Ancestro Común Más Bajo (LCA) de dos nodos u y v en un árbol es el nodo con la profundidad mínima (menor nivel) en el sub-arreglo del Recorrido de Euler (L) comprendido entre la primera aparición de u y la primera aparición de v.

Esto reduce el problema de LCA a un problema de Consulta de Rango Mínimo. (Nota: la justificación es en base 1-indexación)

Tenemos los arreglos

- $EulerLevel = [1, 2, 3, 2, 1, 2, 3, 4, 5, 5, 3, 4, 3, 2, 1]$
- $EulerNodos = [j, i, a, i, j, u, n, o, l, o, n, r, n, u, j]$

1. Identificar las primeras ocurrencias:
   - Posición primera ocurrencia de l: 9.
   - Posición primera ocurrencia de a: 3.
2. Determinar los niveles en el rango [3..9]:
   - $EulerLevel[3..9] = [3, 2, 1, 2, 3, 4, 5]$.
   - $EulerNodos[3..9] = [a, i, j, u, n, o, l]$.
3. Encontrar el Mínimo Nivel:
   - El valor mínimo en la secuencia de niveles [3..9] es 1, el cual ocurre en el índice 5.
4. Identificar el LCA:
   - El nodo en la secuencia E en el índice 5 es j. Por lo tanto, el Ancestro Común Más Bajo entre 'l' y 'a' es j.

Pregunta 2

Pregunta 3

Pregunta 4

Definimos un **grafo de conflicto** $G = (C, E)$, donde:

- E contiene una arista (x, y) si $x, y \in C$ y su suma $x + y$ es un número primo.

Un grafo es **bipartito** si sus vértices pueden ser divididos en dos conjuntos disjuntos A y B tales que cada arista conecta un vértice de A con uno de B.

1. La suma de dos números, $x + y$, es un número primo. Dado que 2 es el único número primo par, y que los números en C son distintos (lo que implica que $x + y \neq 2$ si $x \neq y$ y $x, y > 0$), cualquier suma prima $x + y = P$ debe ser un número primo **impar** ($P > 2$).
2. Para que la suma de dos enteros sea impar, uno debe ser par y el otro debe ser impar.
3. Si definimos A como el conjunto de números impares en C, y B como el conjunto de números pares en C, toda arista $(x, y) \in E$ necesariamente conecta un nodo en A con un nodo en B.

Concluimos que el grafo de conflicto G es **bipartito**.

El objetivo es encontrar el subconjunto máximo de vértices $C' \subseteq C$ tal que no haya aristas entre ningún par de vértices en C'. Este subconjunto es conocido como el **Conjunto Independiente Máximo (MIS)**. El número mínimo de elementos a eliminar es, por definición, $|C| - |MIS|$.

El **Teorema de König** establece una equivalencia fundamental en grafos bipartitos: **la cardinalidad del emparejamiento máximo (MCBM) es igual a la cardinalidad del cubrimiento de vértices mínimo (MVC)**.

$$|MCBM| = |MVC|$$

Además, en cualquier grafo, el Conjunto Independiente Máximo ($|MIS|$) y el Cubrimiento de Vértices Mínimo ($|MVC|$) se relacionan por:

$$|MIS| = |V| - |MVC|$$

Sustituyendo el Teorema de König en esta relación, para nuestro grafo bipartito G:

$$|MIS| = |C| - |MCBM|$$

Dado que buscamos el **mínimo número de números a eliminar** ($|R|$), y $|R| = |C| - |MIS|$, entonces:

$$|R| = |C| - (|C| - |MCBM|) = |MCBM|$$

Por lo tanto, el problema se reduce a calcular la cardinalidad del **Emparejamiento Bipartito Máximo (MCBM)** en el grafo de conflicto G.

Algoritmo

El algoritmo consiste en dos fases principales:

1. Construcción del grafo
2. Cálculo del MCBM.

Fase 1: Construcción del Grafo

1. **Precomputación de Primos:** Se debe determinar si las sumas $x + y$ son primas. Acá se asume por condición del problema que podemos consultar si un número es primo en $O(1)$.
2. **Construcción de Aristas:** Iteramos sobre todos los pares (x, y) donde $x \in A$ (impares) y $y \in B$ (pares). El número total de pares es n^2.
    - Si $x + y$ es primo, agregamos la arista (x, y) a E.
    - El tiempo de construcción es $O(n^2) \times O(1) = O(n^2)$.

Fase 2: Cálculo del Emparejamiento Bipartito Máximo (MCBM)

La cardinalidad del MCBM puede encontrarse usando algoritmos de flujo máximo (Max Flow) como Edmonds-Karp, o usando el algoritmo de Hopcroft-Karp.

El algoritmo de **Hopcroft-Karp** es uno de los métodos más eficientes para encontrar el MCBM, con una complejidad de tiempo de $O(Edges \cdot \sqrt{Nodos})$.

Sustituyendo $Edges = n^2$ y $Nodos = |C| = n$ tenemos $O(n^2 \cdot \sqrt{n})$