# IIC2133 – Estructuras de Datos y Algoritmos

## Examen

Hora inicio: 9:00 del 8 de julio del 2020

Hora máxima de entrega: 15:00 del 8 de julio del 2020

0. Responde esta pregunta en papel y lápiz, incluyendo tu firma al final. Nos reservamos el derecho a no corregir tu prueba según tu respuesta a esta pregunta. Puedes adjuntar esta pregunta a cualquiera de las preguntas que subas a SIDING.

   a. ¿Cuál es tu nombre completo?
   b. ¿Te comprometes a no preguntar ni responder dudas del examen a nadie que no sea parte del cuerpo docente del curso, ya sea de manera directa o indirecta?

***Responde sólo 3 de las 4 preguntas a continuación. Si respondes 4, se escogerá arbitrariamente cuales 3 corregir, y la otra se considerará como no entregada.***

1. Se tiene el siguiente algoritmo de ordenación para un arreglo $\boldsymbol{A}$ de $\boldsymbol{n}$ pares $(\boldsymbol{key}, \boldsymbol{value})$, donde las $\boldsymbol{key}$ son números **naturales**:

$\boldsymbol{IndexSort}(\boldsymbol{A}, \boldsymbol{n})$:

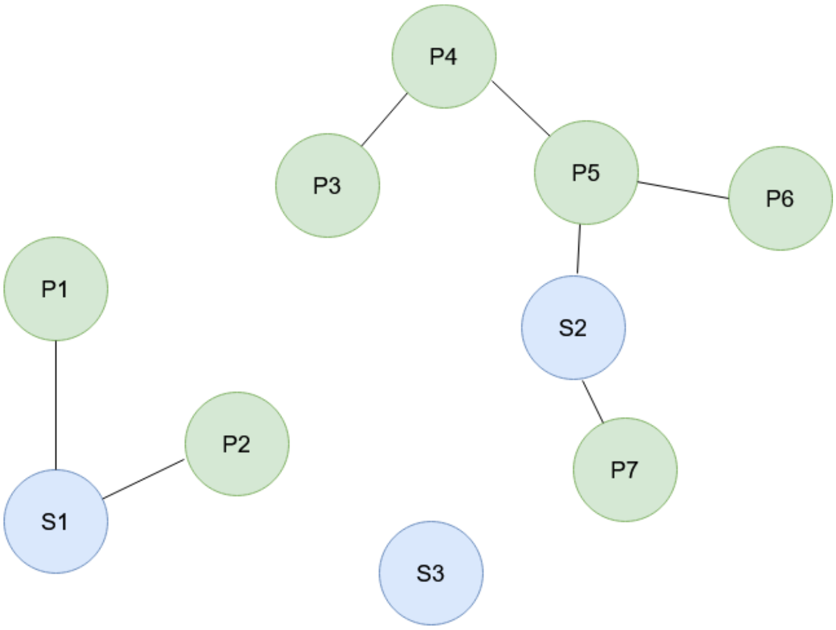
- $\boldsymbol{min} \leftarrow$ el mínimo elemento entre las claves de $\boldsymbol{A}$
- $\boldsymbol{max} \leftarrow$ el máximo elemento entre las claves de $\boldsymbol{A}$
- $\boldsymbol{range} \leftarrow \boldsymbol{max} - \boldsymbol{min} + \mathbf{1}$
- $\boldsymbol{INDEX} \leftarrow$ un arreglo de largo $\boldsymbol{range}$ lleno con ceros
- $\boldsymbol{for}\ \boldsymbol{a}\ \boldsymbol{in}\ \boldsymbol{A}$:
  - Incrementar $\boldsymbol{INDEX}[\boldsymbol{a}.\boldsymbol{key} - \boldsymbol{min}]$ en $\mathbf{1}$
- $\boldsymbol{for}\ \boldsymbol{x}\ \boldsymbol{in}\ \mathbf{1} \ldots \boldsymbol{range} - \mathbf{1}$:
  - $\boldsymbol{INDEX}[\boldsymbol{x}] \leftarrow \boldsymbol{INDEX}[\boldsymbol{x}] + \boldsymbol{INDEX}[\boldsymbol{x} - \mathbf{1}]$
- $\boldsymbol{S} \leftarrow$ un arreglo de largo $\boldsymbol{n}$ para guardar el arreglo ordenado
- $\boldsymbol{for}\ \boldsymbol{i}\ \boldsymbol{in}\ \boldsymbol{n} - \mathbf{1} \ldots \mathbf{0}$:
  - Disminuir $\boldsymbol{INDEX}[\boldsymbol{A}[\boldsymbol{i}].\boldsymbol{key} - \boldsymbol{min}]$ en $\mathbf{1}$
  - $\boldsymbol{index} \leftarrow \boldsymbol{INDEX}[\boldsymbol{A}[\boldsymbol{i}].\boldsymbol{key} - \boldsymbol{min}]$
  - $\boldsymbol{S}[\boldsymbol{index}] \leftarrow \boldsymbol{A}[\boldsymbol{i}]$
- $\boldsymbol{return}\ \boldsymbol{S}$

a) Justifica por qué $\boldsymbol{IndexSort}$ es correcto. No es necesario hacer una demostración formal.
b) Calcula su complejidad. Considera que encontrar el mínimo y el máximo toma $\boldsymbol{O}(\boldsymbol{n})$.
c) Identifica al menos 2 diferencias fundamentales que tiene este algoritmo con los algoritmos vistos en clases.

2. Luego de una traumática experiencia en Ingeniería Comercial, Patrick decidió renunciar y dedicarse a su verdadera pasión: la jardinería. Planificando el sistema de riego para los jardines del tacaño duque Reginald XII, Patrick se encontró con un problema de optimización que le trajo recuerdos:

   Dada una serie de puntos $\boldsymbol{P}$ que deben ser regados, y una serie de puntos $\boldsymbol{S}$ de tomas de agua, se debe disponer de cañerías entre puntos de manera que desde cada punto $\boldsymbol{p} \in \boldsymbol{P}$ exista una **única** ruta mediante cañerías a algún punto $\boldsymbol{s} \in \boldsymbol{S}$. Considerando que el costo de una cañería es proporcional a su largo, se quiere resolver este problema **minimizando** el costo total de las cañerías dispuestas.

   Considera el siguiente ejemplo de una solución:




   Dado $\boldsymbol{S}, \boldsymbol{P}$ y el grafo no dirigido y con costos $\boldsymbol{G}(\boldsymbol{V}, \boldsymbol{E})$, con $\boldsymbol{V} = \boldsymbol{S} \cup \boldsymbol{P}$ y $\boldsymbol{E} = \boldsymbol{V} \times \boldsymbol{V}$; es decir, un grafo completo. Diseña un algoritmo que resuelva este problema en tiempo $\boldsymbol{O}(\boldsymbol{E} \cdot \log \boldsymbol{V})$. Dicho algoritmo puede ser en prosa o en pseudocódigo: evita lenguajes de programación.

3. Respecto a los árboles rojo negro:

   a) Justifica que la rama más larga del árbol tiene a lo más el doble de nodos que la rama más corta. Entiéndase por rama la ruta de la raíz hasta una hoja.

   b) En el algoritmo de inserción estudiado en clases, un nodo recién insertado se pinta de rojo. Con esto, corremos el riesgo de violar la propiedad 3 de árbol rojo-negro (según las diapositivas); y cuando así ocurre, usamos rotaciones y cambios de color para restaurar esa propiedad. En cambio, si pintásemos el nodo de negro, no correríamos este riesgo.
      i) ¿Por qué no pintamos de negro un nodo recién insertado?
      ii) Si lo hiciésemos, ¿qué habría que hacer a continuación para volver a tener un árbol rojo-negro?

   c) Considera un árbol rojo-negro formado mediante la inserción de *n* nodos, siguiendo el algoritmo de inserción estudiado en clase.

      Justifica que si *n* > 1, entonces el árbol tiene al menos un nodo rojo.

4. Se tiene una colección de $\boldsymbol{k}$ jarros de volúmenes $\{\boldsymbol{v_1}, \cdots, \boldsymbol{v_k}\}$ todos **distintos** entre sí, donde el volumen de cada jarro es un número primo de litros.

   Se quiere usar estos jarros para sumar un volumen arbitrario $\boldsymbol{x}$ mediante las acciones de costo 1:

   - $\infty \to \boldsymbol{v_i}$: llenar el jarro $\boldsymbol{i}$ con $\boldsymbol{v_i}$ litros.
   - $\boldsymbol{v_i} \to \boldsymbol{v_j}$: llenar el jarro $\boldsymbol{j}$ con el contenido $\boldsymbol{v} \leq \boldsymbol{v_i}$ del jarro $\boldsymbol{i}$. Así, el jarro $\boldsymbol{i}$ queda con $\boldsymbol{v} - \boldsymbol{v_j}$ litros.

   Y las acciones de costo 0:

   - $\boldsymbol{v_i} \to \emptyset$ : desechar el contenido del jarro $\boldsymbol{i}$
   - $\boldsymbol{v_i} \to \boldsymbol{F}$ : traspasar el contenido del jarro $\boldsymbol{i}$ a un recipiente final $\boldsymbol{F}$

   Considera el siguiente ejemplo de costo 4 con los jarros $\{\mathbf{5}, \mathbf{7}\}$ y $\boldsymbol{x} = \mathbf{4}$:

   - $\infty \to \mathbf{7}$
   - $\mathbf{7} \to \mathbf{5}$
   - $\mathbf{7} \to \boldsymbol{F}$
   - $\mathbf{5} \to \emptyset$
   - $\infty \to \mathbf{7}$
   - $\mathbf{7} \to \mathbf{5}$
   - $\mathbf{7} \to \boldsymbol{F}$

   a) Escribe la ecuación de recurrencia que calcule el costo **mínimo** de resolver este problema.
   b) Mediante diagramas, justifica la utilidad de aplicar programación dinámica a este problema.