

TAREAS Y PROYECTOS

1. Distancia Entre Algoritmos

Utilizar lo visto en clase para implementar un algoritmo en python que permita calcular la distancia entre dos documentos. Los documentos estarán guardados en archivos .txt y se leerán como parametros de ejecución del programa (tarea1.py documento1.txt documento2.txt). El programa debe retornar la distancia entre los dos documentos medida en radianes. Trate de hacer una implementación que sea de rápida ejecución, para ello deberá investigar los tiempos teóricos de ejecución de las funciones a utilizar dentro de python.

2. Heaps

Dado el arreglo de números enteros *nums* se eligen dos índices diferentes *i* y *j* del arreglo. Se debe retornar el valor máximo de $nums[i]-1*(nums[j]-1)$.

Ejemplo:

Entrada: `nums = [3, 4, 5, 2]`

Salida: 12

Explicación: Si se eligen los índices $i = 1$ y $j = 2$ se obtendrá el producto más grande, a saber, $(4 - 1) * (5 - 1) = 12$.

Restricciones:

- $2 \leq \text{nums.length} \leq 500$
- $1 \leq \text{nums}[i] \leq 10^3$

[Fuente](#)

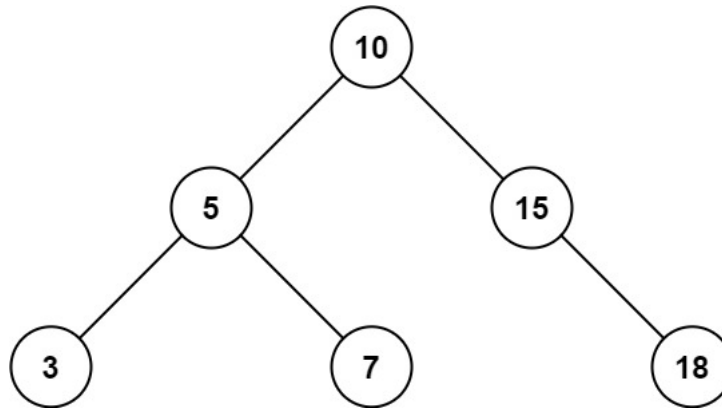
3. Algoritmo Karp–Rabin

Leer el artículo [Geeks for Geeks](#) y ver el vídeo [Youtube](#). Con ello elaborar un documento en LaTeX con tres bullet para el artículo y tres para el vídeo de tres o cuatro líneas que describa las ideas centrales de cada uno.

4. Tarea 4

Dada el nodo *raíz* de un árbol de búsqueda binaria y dos enteros *low* y *high*, retornar el valor de la suma de todos los valores de todos los nodos cuyos valores, inclusive, están en el rango $[low, high]$.

Ejemplo 1:

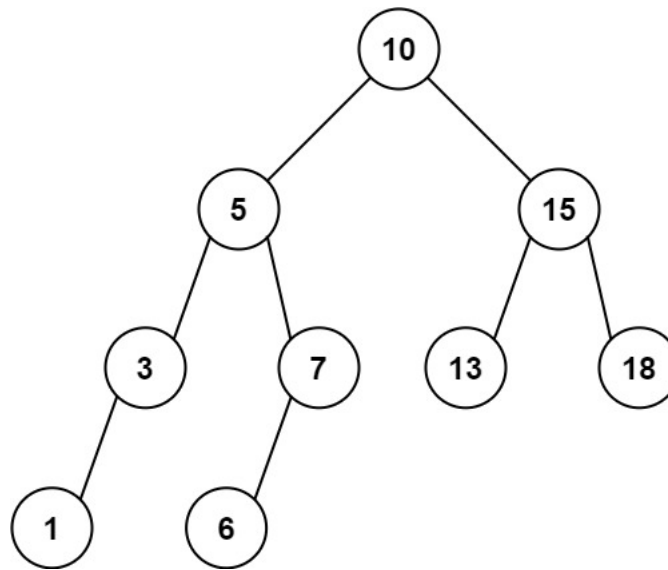


Entrada: root = [10, 5, 15, 3, 7, null, 18], low = 7, high = 15

Salida: 32

Explicación: Nodes 7, 10 and 15 are in the range [7, 15]. $7 + 10 + 15 = 32$.

Ejemplo 2:



Entrada: root = [10, 5, 15, 3, 7, 13, 18, 1, null, 6], low = 6, high = 10

Salida: 23

Explicación: Nodes 6, 7, and 10 are in the range [6, 10]. $6 + 7 + 10 = 23$.

Restricciones:

- El número de nodos en el árbol está en el rango $[1, 2 * 10^4]$.
- $1 \leq \text{Node.val} \leq 10^5$
- $1 \leq \text{low} \leq \text{high} \leq 10^5$
- Todos los Node.val son únicos **unique**.

[LeetCode](#)

5. Hoja de Trabajo

Ver [Youtube](#) y [Youtube 2](#). Con ello elaborar un documento en LaTeX con tres bullet para cada video, de tres o cuatro lineas que describa las ideas centrales de cada uno.

6. Práctica 1

Inscribirse al curso [Study Plan](#) al plan de Estructuras de Datos I. El programa consiste en ir resolviendo problemas diariamente. Su práctica consiste en irlos resolviendo, ingresando sus soluciones en el LeetCode y además crear un repositorio de soluciones en GitHub que deberán subir. Además, deberán elegir uno de los problemas a la semana y hacer su análisis algoritmico (uno diferente cada uno, se ponen de acuerdo entre ustedes cual van a elegir para que no tengamos repetidos), poniendo el código en un documento LaTeX y su respectivo análisis. Al finalizar, deberán presentar el documento PDF y código LaTeX, en el documento PDF deberán poner el enlace al repositorio de Github.

7. Práctica 2

Resolver los problems sets 5 y 6 que se encuentran en [MIT](#). Entregar un solo PDF hecho en LaTeX para estos dos problems sets. En los casos en que pida escribir pseudocódigo, ustedes deberán escribirlo en python. Al final del reporte, deben consignar el repositorio github donde se pueden ver los códigos.

8. Práctica 3

Inscribirse a los cursos [Study Plan](#) a los planes Algorithms I y II . El programa consiste en ir resolviendo problemas diariamente. Su práctica consiste en irlos resolviendo, ingresando sus soluciones en el LeetCode y además crear un repositorio de soluciones en GitHub que deberán subir. Además, deberán elegir uno de los problemas a la semana y hacer su análisis algoritmico (uno diferente cada uno, se ponen de acuerdo entre ustedes cual van a elegir para que no tengamos repetidos), poniendo el código en un documento LaTeX y su respectivo análisis. Al finalizar, deberán presentar el documento PDF y código LaTeX, en el documento PDF deberán poner el enlace al repositorio de Github.