



PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE  
ESCUELA DE INGENIERIA  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACION

Diseño y Análisis de Algoritmos - IIC2283 - 2<sup>do</sup> semestre - 2024

# Tarea 2

## Programación Dinámica y Algoritmos Codiciosos

Publicación : Viernes 18 de octubre  
Github Classroom : [https://classroom.github.com/a/64aK0V2\\_](https://classroom.github.com/a/64aK0V2_)  
Entrega : Jueves 31 de octubre

### Indicaciones

- La tarea es estrictamente individual.
- La solución debe ser entregada en los archivos `t2p1.py` y `t2p2.py` del repositorio privado asignado mediante GitHub Classroom para esta tarea. Se revisará el último *commit* subido antes de la entrega al repositorio. Se usará Python 3.10.X para la revisión.
- El *input* para el programa debe ser obtenido desde *standard input*. El *output* debe ser entregado mediante *standard output*.
- La corrección se realizará mediante *tests* automatizados acordes al formato de *input* y *output* especificado. Cada *test* tendrá un timeout según lo que se especifica como tiempo esperado.
- Un *test* se considerará **reprobado** en caso de que 1) dado el *input* el *output* sea incorrecto, 2) exista un error de *runtime* durante su ejecución, o 3) el *timeout* se cumpla durante su ejecución. En otro caso, el *test* se considerará **aprobado**.
- No se permite el uso de librerías externas a la librería estándar de Python a *priori*. Consultar en las [issues del repositorio oficial del curso](#) en caso de requerir una.
- A 5 días de la fecha final del plazo, se publicará un test de gran tamaño para cada parte, siendo el único test que publicaremos. La habilidad de generar tests es parte de la evaluación.
- Para esta tarea aplica la política de atrasos descrita en el programa del curso.
- **NO** se garantiza la existencia de una nueva instancia de recorreción. Deben asegurarse de entregar el output **solo** por *standard output* y **no** imprimir el tiempo.

## 1. Bosque superexponencial (60 %)

Charlie se encuentra estudiando la similitud entre árboles de su jardín, por lo que quiere saber cuantos árboles similares a un árbol de su casa existen. Para esto, representó su árbol favorito mediante nodos numerados de forma secuencial (entre 1 y  $N$ ) y planteó una categoría por cada grado de similitud  $X_i$  ( $0 \leq i \leq N - 1$ ). La categoría  $X_i$  contiene todos los árboles de  $N$  nodos que poseen exactamente  $i$  aristas en común al árbol favorito de Charlie. Una arista se considera común si conecta los mismos nodos  $a$  y  $b$  en ambos. ¿Serías capaz de decir cuantos árboles hay en cada categoría?

### Input

La primera línea consiste en un entero  $N$  ( $2 \leq N \leq 800$ ), indicando cuantos nodos tiene el árbol de Charlie. A continuación se entregan  $N - 1$  líneas con enteros  $a$  y  $b$  ( $1 \leq a < b \leq 800$ ) indicando que los nodos  $a$  y  $b$  están conectados.

Está garantizado que el input es un árbol válido.

### Output

$N$  líneas, una por cada  $X_i$  ( $0 \leq i \leq N - 1$ ) indicando la cantidad de árboles en dicha categoría. Para no afectar a la complejidad del problema, deberá entregar sus respuestas en módulo 2147483647.

### Tiempo límite

Se espera que la solución se ejecute en un tiempo **menor o igual a 2 segundos** para cualquier instancia de input según las restricciones dadas.

### Complejidad esperada

Se espera que la solución posea una complejidad de  $O(N^2)$ .

### Hint

Use programación dinámica. La [formula de Cayley](#) indica que dado un numero entero  $N > 1$ , la cantidad de arboles posibles que se pueden construir con  $N$  nodos numerados de forma secuencial es  $N^{N-2}$ , por ende, la suma de sus respuestas debería dar exactamente el numero dado por la formula anterior. La pagina referenciada incluye un dibujo que les servira para contar los arboles para  $2 \leq N \leq 4$ .

## Ejemplos

Input de ejemplo 1
3 1 2 1 3
Output de ejemplo 1
0 2 1

Input de ejemplo 2
4 1 4 2 4 3 4
Output de ejemplo 2
0 9 6 1

Input de ejemplo 3
5 1 2 2 3 3 4 3 5
Output de ejemplo 3
8 50 52 14 1

## 2. Problemas de luminarias (40 %)

La municipalidad de Tuki ha realizado un estudio sobre las luminarias de las principales avenidas de la comuna y se dio cuenta de que los gastos asociados a estos no son sólo excesivos, sino que también innecesarios en algunas partes. Esto es porque existen sectores donde hay tantos focos juntos que, en la práctica, tener sólo uno es igualmente efectivo. Por otro lado, pese a tener lugares con exceso de iluminación, también hay sectores que no tienen los niveles mínimos aceptables de luz. Tras un análisis, se han propuesto eliminar la mayor cantidad de faroles innecesarios, manteniendo toda la avenida iluminada, además de reportar cuándo hay lugares a oscuras.

### Input

Cada test consistirá de una cantidad desconocida  $T$  ( $1 \leq T \leq 30$ ) de sub tests, cada uno de estos se verán de la siguiente forma. La primera línea poseerá 2 enteros  $C$  ( $0 \leq C \leq 10^{11}$ ) y  $F$  ( $1 \leq F \leq 10^4$ ), indicando el largo de una avenida (la cual se extiende en el intervalo  $[0, C]$ ) y cuántos faroles tiene inicialmente, luego existirán  $F$  líneas con los enteros  $p_i$  y  $a_i$  ( $0 \leq p_i, a_i \leq 10^{11}$ ), indicando que hay un farol en el punto  $p_i$  y que éste proporciona un nivel aceptable de luz en un radio  $a_i$  alrededor de él. El fin de los tests se indicará con dos números ceros.

### Output

Por cada sub test, se deberá imprimir una línea con un entero que indica la cantidad máxima de faroles que pueden ser eliminados, manteniendo toda la avenida iluminada. En caso de existir algún punto sin niveles aceptables de luz, la respuesta deberá ser  $-1$ .

### Tiempo límite

Se espera que la solución se ejecute en un tiempo **menor o igual a 1 segundo** para cualquier instancia de input según las restricciones dadas.

### Complejidad esperada

Se espera que la solución posea una complejidad por consulta de  $O(T \cdot F \cdot \log(F))$ .

### Hint

Use un algoritmo *greedy*.

## Ejemplo

Input de ejemplo 1
13 3 1 8 9 0 13 4 0 0
Output de ejemplo 1
1

Input de ejemplo 2
7 2 2 2 4 3 9 4 3 3 8 0 2 2 9 1 0 0
Output de ejemplo 2
0 -1

## Política de integridad académica

Los/as estudiantes de la Escuela de Ingeniería de la Pontificia Universidad Católica de Chile deben mantener un comportamiento acorde a la Declaración de Principios de la Universidad. En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los/as estudiantes que incurran en este tipo de acciones se exponen a un Procedimiento Sumario. Es responsabilidad de cada estudiante conocer y respetar el documento sobre Integridad Académica publicado por la Dirección de Docencia de la Escuela de Ingeniería.

Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica. Todo trabajo presentado por un/a estudiante para los efectos de la evaluación de un curso debe ser hecho individualmente por el/la estudiante, sin apoyo en material de terceros. Por “trabajo” se entiende en general las interrogaciones escritas, las tareas de programación u otras, los trabajos de laboratorio, los proyectos, el examen, entre otros.

En particular, si un/a estudiante copia un trabajo, o si a un/a estudiante se le prueba que compró o intentó comprar un trabajo, obtendrá nota final 1.1 en el curso y se solicitará a la Dirección de Pregrado de la Escuela de Ingeniería que no le permita retirar el curso de la carga académica semestral.

Por “copia” se entiende incluir en el trabajo presentado como propio, partes hechas por otra persona. En caso de que corresponda a “copia” a otros estudiantes, la sanción anterior se aplicará a todos los involucrados. En todos los casos, se informará a la Dirección de Pregrado de la Escuela de Ingeniería para que tome sanciones adicionales si lo estima conveniente.

También se entiende por copia extraer contenido sin modificarlo sustancialmente desde fuentes digitales como Wikipedia o mediante el uso de asistentes inteligentes como ChatGPT o Copilot. Se entiende que una modificación sustancial involucra el análisis crítico de la información extraída y en consecuencia todas las modificaciones y mejoras que de este análisis se desprendan. Cualquiera sea el caso, el uso de fuentes bibliográficas, digitales o asistentes debe declararse de forma explícita, y debe indicarse cómo el/la estudiante mejoró la información extraída para cumplir con los objetivos de la actividad evaluativa.

Obviamente, está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la referencia correspondiente.

Lo anterior se entiende como complemento al Reglamento del Estudiante de la Pontificia Universidad Católica de Chile (<https://registrosacademicos.uc.cl/reglamentos/estudiantiles/>). Por ello, es posible pedir a la Universidad la aplicación de sanciones adicionales especificadas en dicho reglamento.

## Compromiso del Código de Honor

Este curso suscribe el Código de Honor establecido por la Universidad, el que es vinculante. Todo trabajo evaluado en este curso debe ser propio. En caso de que exista colaboración permitida con otros/as estudiantes, el trabajo deberá referenciar y atribuir correctamente dicha contribución a quien corresponda. Como estudiante es un deber conocer el Código de Honor (<https://www.uc.cl/codigo-de-honor/>)