



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

Diseño y Análisis de Algoritmos - IIC2283 - 2^{do} semestre - 2025

Tarea 1

Dividir y conquistar y Programación Dinámica

Publicación : Miercoles 10 de septiembre
Github Classroom : <https://classroom.github.com/a/HqnWoDd0>
Entrega : Miercoles 1 de octubre

Indicaciones

- La tarea es estrictamente individual.
- La solución debe ser entregada en los archivos `t1p1.py` y `t1p2.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. Concurso de cueca (50 %)

En un gran edificio donde cada celda es una habitación dedicada a bailar cueca (el baile tradicional chileno), las parejas de bailarines se preparan para un importante concurso. El edificio tiene n pisos. Cada piso tiene m habitaciones con un solo pasillo que las conecta. Las habitaciones están numeradas del 1 al m a lo largo del pasillo, y todas las habitaciones con el mismo número en diferentes pisos se encuentran exactamente una encima de la otra. Por lo tanto, el edificio puede representarse como una grilla de altura n y ancho m . Podemos denotar las habitaciones con pares de enteros (i, j) donde i es el piso y j el número de habitación en ese piso.

Cada pareja puede moverse por el pasillo de cada piso, usar escaleras o ascensores. Cada escalera o ascensor ocupa todas las habitaciones $(1, x), (2, x), \dots, (n, x)$ para algún x entre 1 y m . Todas las habitaciones no ocupadas por escaleras o ascensores están destinadas a bailar.

Se necesita una unidad de tiempo para moverse entre habitaciones vecinas en el mismo piso o para subir o bajar un piso usando las escaleras. Se necesita una unidad de tiempo para moverse hasta v pisos en cualquier dirección usando un ascensor. No hay tiempo de espera por un ascensor, y el tiempo para entrar o salir de él es despreciable.

Actualmente, una pareja se encuentra en la habitación $(x1, y1)$ y quiere llegar a su habitación correspondiente para el concurso en $(x2, y2)$ en el menor tiempo posible.

Se deben procesar q consultas, cada una preguntando: "¿Cuál es el tiempo mínimo necesario para que la pareja llegue desde $(x1, y1)$ hasta $(x2, y2)$?"

Input

La primera línea contiene cinco enteros: n, m, c_e, c_a, v , el número de pisos, el número de habitaciones por piso, la cantidad de escaleras, la cantidad de ascensores y la velocidad máxima de un ascensor, respectivamente ($2 \leq n, m \leq 10^8, 0 \leq c_e, c_a \leq 10^5, 1 \leq c_e + c_a \leq m - 1, 1 \leq v \leq n - 1$).

La segunda línea contiene c_e enteros e_i , en orden creciente, indicando las posiciones de las escaleras ($1 \leq e_i \leq m$). Si $c_e = 0$, la segunda línea está vacía.

La tercera línea contiene c_a enteros a_i , en orden creciente, indicando las posiciones de los ascensores en el mismo formato. Se asegura que todos los enteros e_i y a_i son distintos.

La cuarta línea contiene un solo entero q , el número de consultas ($1 \leq q \leq 10^5$).

Las siguientes q líneas describen las consultas. Cada línea contiene cuatro enteros: $x1, y1, x2, y2$: las coordenadas de la habitación inicial y final de la pareja. Se garantiza que las habitaciones inicial y final son distintas, y que no hay una escalera o ascensor en la columna $y1$ ni $y2$ ($1 \leq x1, x2 \leq n, 1 \leq y1, y2 \leq m$).

Output

Imprimir q enteros, uno por línea — el tiempo mínimo para que cada pareja llegue a su habitación correspondiente.

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 $O(q \log \max(c_e, c_a))$.

Hint

Use búsqueda binaria. piense cómo se vería el camino óptimo que lleva desde $(x1, y1)$ a $(x2, y2)$ y busque una manera de encontrarlo de forma rápida.

Ejemplos

Input de ejemplo 1
5 6 1 1 3 2 5 3 1 1 5 6 1 3 5 4 3 3 5 3
Output de ejemplo 1
7 5 4

Input de ejemplo 2
4 4 1 0 1 4 5 1 1 2 2 1 3 2 2 3 3 4 3 3 2 2 2 1 2 2 3
Output de ejemplo 2
6 4 3 5 4

2. Festival de empanadas (50 %)

El gran Festival de la Empanada ha llegado, y tú, como chef jefe, tienes la misión de formar los equipos más talentosos para maximizar las ganancias. Se han inscrito n participantes, numerados del 0 al $n - 1$, donde n es un múltiplo de 3. Debes organizar a estos participantes en equipos de tres. Cada persona debe estar en un equipo y no puede formar parte de más de uno.

El puntaje total del festival es la suma de los puntajes de cada equipo. Para maximizar las ganancias, debes elegir la composición de cada equipo de forma óptima. Una vez que un equipo de tres personas se ha formado, ellos decidirán internamente la mejor manera de colaborar para obtener el mayor puntaje posible.

Cada equipo puede preparar un tipo de empanada distinto, y el puntaje que obtienen depende de cómo colaboran sus miembros. La compatibilidad entre dos chefs se mide por el sabor único que sus empanadas adquieren al combinar sus recetas. Esta compatibilidad está dada por una matriz de nn celdas, donde $coop_{i,j}$ representa la sinergia entre el chef i y el chef j .

Además, cada chef tiene una habilidad individual para crear una empanada clásica por sí mismo. Esto se representa en un arreglo a de tamaño n , donde a_i es la habilidad individual del chef i .

Cuando un equipo de tres chefs, digamos i, j, k , se forma, tienen las siguientes opciones para decidir su estrategia y maximizar su propio puntaje:

- Si uno solo de los chefs participa activamente: El puntaje del equipo será la habilidad individual de esa persona. Por ejemplo, si solo participa el chef i , el puntaje es a_i .
- Si dos de los chefs participan activamente: El puntaje del equipo será la compatibilidad entre ambos. Por ejemplo, si participan los chefs i y j , el puntaje es $coop_{i,j}$.
- Si los tres chefs participan activamente: El puntaje del equipo es el producto de todas las compatibilidades entre ellos. Si los tres chefs i, j, k participan, el puntaje es $coop_{i,j} \cdot coop_{i,k} \cdot coop_{j,k}$.

Tu tarea es formar los equipos de tres personas de manera que la suma total de puntajes de todos los equipos sea la máxima posible. Considera que, una vez formados, cada equipo elegirá su estrategia de colaboración para obtener el mayor puntaje individual.

Input

La entrada consiste en varias líneas. La primera con un entero n , la cantidad de personas, se asegura que n es un múltiplo de 3 ($3 \leq n \leq 18$).

Las siguientes n líneas contienen la descripción de la compatibilidad entre participantes.

La i -ésima fila contiene n enteros, $coop_{i,j}$, la compatibilidad entre la i -ésima y j -ésima persona ($0 \leq |coop_{i,j}| \leq 10^5$). Se asegura que $coop_{i,j} = coop_{j,i}$ y que $coop_{i,i} = 0$ para todo i .

La última línea contiene n enteros, a_i , la habilidad individual de la persona i ($1 \leq a_i \leq 10^9$).

Output

La salida debe contener un único entero correspondiente a la máxima cantidad de puntaje que puede conseguir el equipo chileno si se eligen los equipos de forma óptima.

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 $O(2^n \cdot n^2)$.

Adicional

Soluciones que no utilicen programación dinámica no serán consideradas y serán evaluadas con la nota mínima

Ejemplo

Input de ejemplo 1
3 0 -8 -3 -8 0 -4 -3 -4 0 43 47 50
Output de ejemplo 1
50

Input de ejemplo 2
6 0 -2 -5 -7 0 -3 -2 0 0 6 -3 -5 -5 0 0 -1 -4 6 -7 6 -1 0 -5 5 0 -3 -4 -5 0 6 -3 -5 6 5 6 0 13 7 37 9 16 16
Output de ejemplo 2
180

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/>)