ISIS 1105 Diseño y Análisis de Algoritmos
Semestre 2025-10. Proyecto – PARTE 1
Entrega: viernes, 14 de marzo de 2025 6:00PM (por Bloque Neon)

0 OBJETIVOS

- Diseñar soluciones computacionales para problemas.
- Estimar costos de las soluciones planteadas.
- Implementar soluciones.

Se premiarán las mejores soluciones y se castigarán las peores, en cuanto a optimización y eficiencia en tiempo y espacio.

1 CONDICIONES GENERALES

El proyecto se divide en tres partes independientes entre sí. Este documento describe la PARTE I. Cada parte contiene un problema a resolver mediante soluciones implementadas en `Java` o `Python`.

Para cada problema se pide:
- Descripción de la solución.
- Análisis temporal y espacial.
- Una implementación en Java o Python

2 DESCRIPCIÓN DEL PROBLEMA

Usted y su grupo han sido seleccionados para participar en el Juego del Algormar, un desafío legendario en el que solo el equipo más astuto y estratégico logrará sobrevivir hasta el final. El primer evento, conocido como "El menor peso te salva", ha comenzado. Todos los equipos han sido alineados en una extensa plataforma flotante sobre un abismo desconocido. Cada jugador tiene un peso energético p_i, el cual es un valor que se le asigna al jugador una vez sube a la plataforma.

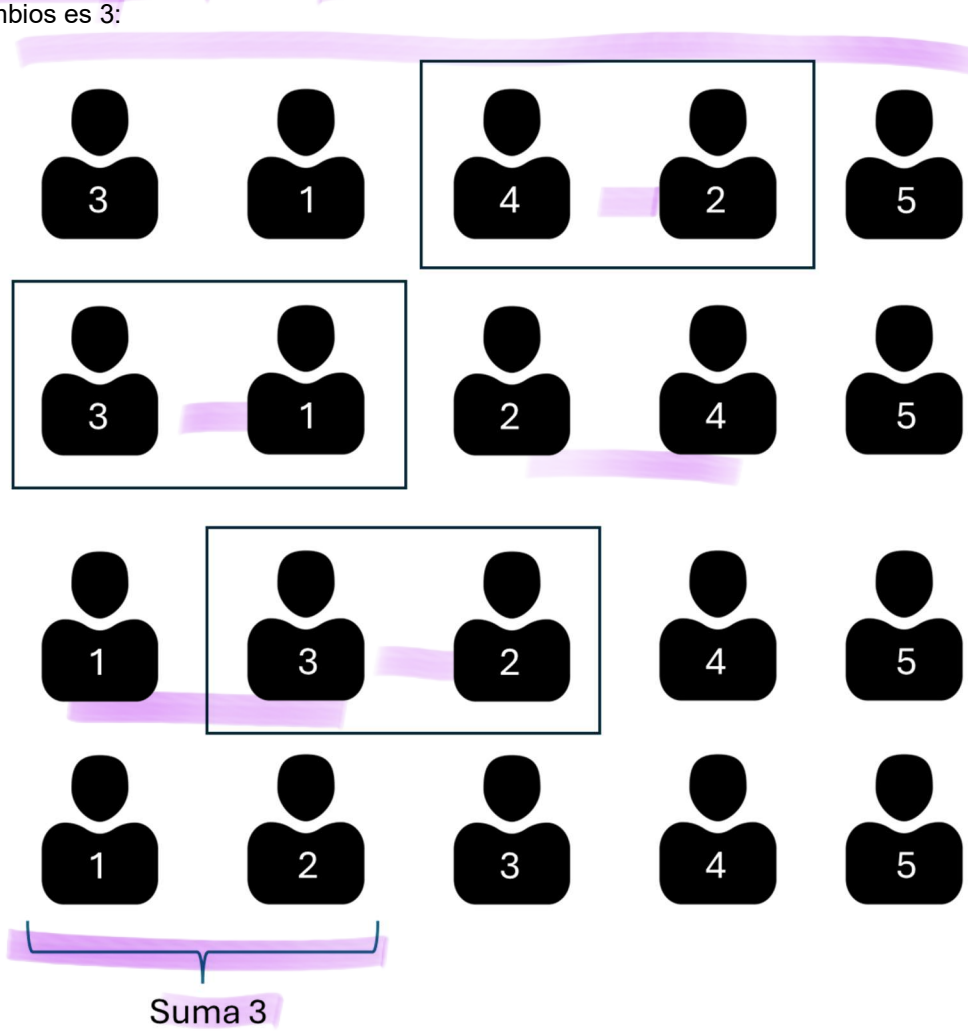
Para superar este reto, su equipo debe asegurarse de que los j primeros compañeros de su fila tengan el menor peso energético posible, evitando que la plataforma se desequilibre y terminen el vacío. Las reglas del juego les permiten realizar hasta m intercambios entre jugadores consecutivos (adyacentes) para mejorar la alineación lo más que se pueda.

Problema

Determinar cuál es el peso energético total mínimo de los primeros j jugadores en la línea que se puede lograr al realizar no más de m intercambios de jugadores consecutivos.

Suponga que su equipo es de $n = 5$ jugadores, $j = 2$ (compañeros en la fila que cuentan en el acumulado de peso), y $m = 3$ (número de intercambios admitidos). Al inicializar el juego los pesos energéticos quedan asignados de la siguiente manera:

El menor peso energético de los j primeros elementos en la fila que se puede obtener en m intercambios es 3:



3 ENTRADA Y SALIDA DE DATOS

En todas las soluciones que se presenten, la lectura de los datos de entrada se hace por la entrada estándar; así mismo, la escritura de los resultados se hace por la salida estándar.

Puede suponer que ninguna línea de entrada tiene espacios al principio o al final, y que los datos que se listan en cada línea están separados por exactamente un espacio.

A continuación, se establecen parámetros que definen su tamaño y formato de lectura de los datos, tanto de entrada como de salida.

Descripción de la entrada

La primera línea de entrada especifica el número de casos de prueba que contiene el archivo. El programa debe terminar su ejecución, una vez termine de resolver la cantidad de casos de prueba dados por este número.

Cada caso de prueba esta descrito en una línea. La línea comienza con 3 números naturales que representan n, j, y m. Seguido se presentan n enteros que representan los valores q_i. Restricciones de tamaño:

$1 \leq n \leq 10^3, 1 \leq j \leq n \leq 10^3, 1 \leq m \leq 10^4, 1 \leq q_i \leq 10^5$

Descripción de la salida

Para cada caso de prueba, la salida deben ser una sola línea con el mínimo de energía posible.

Ejemplo de entrada / salida

Entrada	Salida
5 5 2 3 3 1 4 2 5 8 3 6 57 43 31 21 13 1 7 3 13 7 20 57 27 13 91 73 1 13 1 43 21 31 3 7 17 2 2 43 81 103 13 27 61 43 31 21 13 1 7 1 3 91 73 57 23 11 19 127 103 1 23 81 43 61 153 181 47 7 3 27 91 43 57 21 1 73 13 13 1 31	3 65 79 56 463

Nota: Se van a diseñar casos de prueba para valores de n, j, m mucho más grandes y dentro de los valores establecidos en el enunciado. Los casos mostrados en este documento son demostrativos de la estructura de entrada/salida esperada.

5 ENTREGABLES

El proyecto puede desarrollarse por grupos de hasta dos estudiantes de la misma sección. La entrega se hace por bloque neon (una sola entrega por grupo de trabajo).

El grupo debe entregar, por bloque neon, un archivo de nombre `proyectoDalgoP1.zip`. Este archivo es una carpeta de nombre `proyectoDalgoP1`, comprimida en formato `.zip`, dentro de la cual hay archivos fuente de soluciones propuestas y archivos que documentan cada una de las soluciones.

5.1 Archivos fuente de soluciones propuestas

Todos los programas implementados en *Java* o en Python

Para el problema:

- Entregar un archivo de código fuente en *Java* (`.java`) o python (`.py`) con su código fuente de la solución que se presenta.
- Incluir como encabezado de cada archivo fuente un comentario que identifique el (los) autor(es) de la solución.
- Denominar `ProblemaP1.java` o `ProblemaP1.py` el archivo de la solución que se presente.

Nótese que, si bien puede utilizarse un *IDE* como *Eclipse* o Spyder durante el desarrollo del proyecto, la entrega requiere incluir solo un archivo por cada solución. El archivo debe poderse compilar y ejecutar independientemente (sin depender de ninguna estructura de directorios, librerías no estándar, etc.).

5.2 Archivos que documentan la solución propuesta

La solución al problema debe acompañarse de un archivo de máximo 3 páginas que la documente, con extensión `.pdf`. El nombre del archivo debe ser el mismo del código correspondiente (`ProblemaP1.pdf`).

Un archivo de documentación debe contener los siguientes elementos:

0 Identificación
Nombre de autor(es)

1 Algoritmo de solución
Explicación del algoritmo elegido. Si hubo alternativas de implantación diferentes, explicar por qué se escogió la que se implementó. Generar al menos una gráfica que apoye la explicación del algoritmo implementado. No se debe copiar y pegar código fuente como parte de la explicación del algoritmo.

2 Análisis de complejidades espacial y temporal
Cálculo de complejidades y explicación de estas.

La nota del informe corresponde a un 50% de la nota total de la entrega del proyecto, solamente si se entrega el código fuente de la solución implementada. En caso de no entregar el código fuente, no se hará evaluación del informe y la nota del proyecto será cero.

Además de la pertinencia del texto como explicación de la solución implementada, se evaluará la calidad en la redacción del texto y en el diseño de las gráficas. Se evaluará también que la explicación del algoritmo integre los conceptos relacionados con las técnicas de diseño de algoritmos cubiertas en el curso.

Téngase en cuenta que los análisis de 2 tienen sentido en la medida que la explicación de 1 sea clara y correcta. No se está exigiendo formalismo a ultranza, pero sí que, como aplicación de lo estudiado en el curso, se pueda describir un algoritmo de manera correcta y comprensible.

5.3 Consideraciones sobre la entrega

Lea bien las recomendaciones de entrada/salida. Su proyecto será evaluado, en gran parte, por una máquina: si en la ejecución del programa el formato de salida no coincide perfectamente con lo requerido, la calificación de la ejecución será cero, sin importar la cantidad de código que haya desarrollado.

La forma en que se presenta la documentación debe respetarse, aunque su evaluación no sea tan automática como la del software. Hay que nombrar los archivos como se espera que se nombren, comprimirlos como se pide que se compriman, etc. Cualquier desviación en cuanto a lo que se pide produce deméritos en la calificación final.

El software se evalúa desde la linea de comandos de Linux. No hay problema en desarrollar en cualquier otro sistema (Mac, Unix, ...) siempre que se produzca código estándar en java o python. Cada problema se debe resolver en un (1) archivo.