ISIS 1105 Diseño y Análisis de Algoritmos
Semestre 2023-20. Proyecto – PARTE 1
Entrega: viernes, septiembre 29 de 2023 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 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

A Ranas salta piedras

Existen unas ranas en Colombia que no les gusta el agua y se la pasan saltando de piedra en piedra. Suponga un estanque lineal donde hay *p* piedras y donde se posan *r* ranas. En una piedra solo se puede posar una rana. Siempre hay al menos una rana posada sobre una de la *p* piedras y siempre hay al menos una piedra libre.

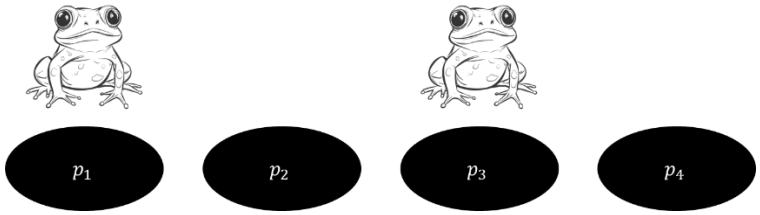
En un momento particular, solo una de las ranas se puede mover siempre y cuando una piedra adyacente este libre. Además de esto, cada rana solamente se mueve en una de las dos direcciones posibles.

Problema

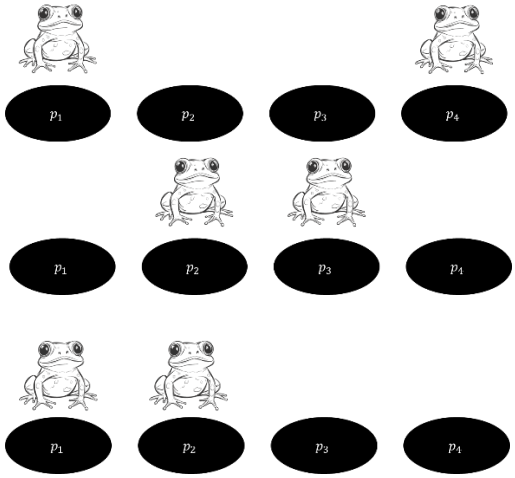
Dada una disposición inicial de ranas y piedras que cumplen el enunciado, determinar cuántas diferentes disposiciones de ranas y piedras diferentes existen luego de realizar *m* movimientos.

Ejemplo 1:

Dado $p=4$, $r=2$, $m=1$ y la distribución de ranas de la siguiente figura:

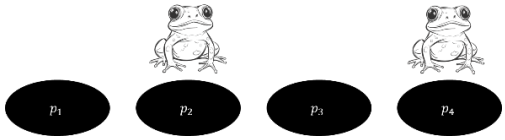


La salida en este ejemplo es 3, que corresponde a las siguentes configuraciones:



Ejemplo 2:

Dado $p=4$, $r=2$, $m=2$ y la distribución de ranas del ejemplo anterior, la salida esperada ahora es 1 que corresponde a la siguiente disposición:



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 está representado por una línea con la especificación de p, r, m y la disposición de ranas y piedras:

$2 \leq p \leq 10^3$

$1 \leq r \leq p - 1$

$1 \leq m \leq p$

La disposición inicial de ranas y piedras será un string de p caracteres. Un carácter “p” implica una piedra vacía y el carácter “r” implica una piedra con rana.

Descripción de la salida

Para cada caso de prueba, imprimir el número de disposiciones diferentes de ranas y piedras luego de los m movimientos.

Nota importante: Como se pueden obtener números muy grandes, para evitar desbordar la capacidad de números enteros, a cada suma que realice para calcular el valor objetivo se le debe aplicar la operación módulo 998244353.

Ejemplo de entrada / salida

Para los ejemplos descritos arriba:

Entrada	Salida
2 4 2 1 rprp 4 2 2 rprp	3 1

Ejemplo con tres casos de prueba:

Entrada	Salida
3 3 1 1 prp 5 4 2 prrrr 9 2 6 ppprpppr	2 1 3

Nota: Se van a diseñar casos de prueba para valores de $p, r,$ y m muchos 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.

4 COMPRENSIÓN DE PROBLEMAS ALGORITMICOS

A continuación, se presentan un conjunto de escenarios hipotéticos que cambian el problema original. Para cada escenario debe contestar[1]: (i) que nuevos retos presupone este nuevo escenario -si aplica-?, y (ii) que cambios -si aplica- le tendría que realizar a su solución para que se adapte a este nuevo escenario?

ESCENARIO 1: Las ranas pueden devolverse

ESCENARIO 2: Todas las ranas deben ir en la misma dirección

Nota: Los escenarios son independientes entre sí.

5 ENTREGABLES

El proyecto puede desarrollarse por grupos de hasta tres 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.).

[1] NO tiene que implementar la solución a estos escenarios, el propósito es meramente analítico.

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)
Identificación 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. Argumentar si el algoritmo planteado resuelve perfectamente el problema.

2 Análisis de complejidades espacial y temporal
Cálculo de complejidades y explicación de estas. Debe realizarse un análisis para cada solución entregada.

3 Respuestas a los escenarios de comprensión de problemas algorítmicos.
Respuesta a las preguntas establecidas en cada escenario. NO tiene que implementar la solución a estos escenarios, el propósito es meramente analítico.

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.