Maestría en Ciencias con Especialidad en Computación y Matemáticas Industriales
Proceso de admisión 2022
Centro de Investigación en Matemáticas (CIMAT), A.C.
Examen de programación
(Tiempo: 3 horas)

Nombre: ______________________________

Fecha: ______________________________

Instrucciones:

- Escriba lo más claro posible para que al digitalizar las respuestas sean legibles.
- Resuelva cada problema en hojas independientes.
- Escriba su nombre en cada hoja.
- No se debería de pasar más de 30 minutos en cada problema.
- Escriba el código lo más claramente posible y especifique el lenguaje/pseudo-código utilizado.
- Si requiere funciones estándar como determinar el máximo o mínimo de dos valores (max(a, b), min(a,b)), ordenar (sort), determinar el tamaño de una cadena de caracteres (strlen), obtener el valor absoluto (abs), raíz cuadrada (sqrt) o logaritmos (log) puede utilizarlas sin implementarlas. Será necesario que implemente cualquier función diferente a las mencionadas anteriormente.

Problema 1 [2 punto]

La mediana de un conjunto finito de valores es aquel valor que divide al conjunto en dos partes iguales, de forma que el número de valores mayor o igual a la mediana es igual al número de valores menores o igual a estos. Por ejemplo, la mediana del conjunto $\{7, 3, 5, 4, 1\}$ es 4 porque 2 valores de este conjunto $(5, 7)$ son superiores a 4 y dos valores $(1, 3)$ son inferiores a 4. Cuando el número de valores en el conjunto es par, no existe un solo valor medio, si no que existe dos valores medios. En tal caso, la mediana es el promedio de éstos. Escriba una funcion que calcule la mediana de un conjunto $\{x_1, \ldots, x_n\}$ de números reales.

Ejemplo 1: La ejecución de la función con el siguiente arreglo de entrada:

$$\{-2, 10, 1, 3, 4\},$$

se debe devolver 3.

Ejemplo 2: La ejecución de la función con el siguiente arreglo de entrada:

$$\{-2, 10, 1, 3, 4, 6\},$$

se debe devolver 3.5 (el promedio de 3 y 4).

Solution: Ordenar el arreglo utilizando una función *sort*. Llamemos N al número de elementos en el arreglo. A través de un if procesar los dos siguientes casos. Si N es par, regresar el promedio de los elementos del arreglo en las posiciones $N/2$ y $N/2+1$. Si N es impar, regresar el elemento del arreglo en la posición $\lceil N/2 \rceil$.

Problema 2 [2 punto]

Sea A una matriz $N \times N$ de coeficientes enteros (positivos o negativos). Escribir una función que determine cuál es la máxima suma de coeficientes que se puede obtener entre todos las sub-matrices de elementos contiguos de A. Recordamos que una sub-matriz de una matriz es la matriz que se puede formar seleccionando cualquier sub-conjunto de filas y columnas contiguas de la matriz original.

Ejemplo: Ante la ejecución de la función con

$$A = \begin{pmatrix} 1 & -10 & 3 \\ -1 & 1 & 4 \\ 0 & 2 & 7 \end{pmatrix}$$

se regresará 14, que se obtiene con la sub-matriz $\begin{pmatrix} 1 & 4 \\ 2 & 7 \end{pmatrix}$.

Solution: Se deben explorar todas las sub-matrices con elementos contiguos de A que tengan tamaño $N_i \times N_j$, con $N_i, N_j \in \{1, 2, ..., N\}$. Esto se puede realizar con varios ciclos anidados, unos que exploren todos los posibles tamaños de sub-matrices $N_i \times N_j$, y otros que accedan a los elementos de la matriz A haciendo los corrimientos adecuados a forma de "ventanas"que se deslizan. Para cada una de esas matrices hay que calcular la suma de sus coeficientes y compararla con la suma S más grande hasta ahora encontrada. S se puede inicializar con el primer coeficiente de la matriz A.

Problema 3 [1.5 punto]

¿Qué calcula la función siguiente para x un flotante y n un entero ≥ 0?

```
def secret(x,n):
    if n==0:
        return 1
    else:
        r = secret(x,n//2)
        if n%2==0:
            return r*r
        else:
            return x*r*r
```

En este código Python, el operador "//´ es el de división entera, y el operador "%" es el de resto por división entera.

Sugerencia: Empezar por ver qué pasa cuando n es una potencia de 2, por ejemplo $n = 4$.

Solution: Es una función recursiva tal que $f_n = (f_{\frac{n}{2}})^2$ si n par, $f_n = x \times (f_{\frac{n}{2}})^2$ si n impar, y $f_0 = 1$. Por recurrencia sobre n, mostramos que cacula la potencia n de x, es decir x^n.

Problema 4 [1.5 punto]

Escribir una función que tome dos cadenas de caracteres s_L y s_C de entrada, con s_C más corta que s_L, y que determine el número de ocurrencias de s_C dentro de s_L.

Ejemplo: Ante la ejecución de la función con las cadenas

"TATATOTA" y "TA"

se regresará 3.

Solution: El algoritmo más simple es un doble ciclo: consiste en recorrer la cadena larga carácter por carácter. En cada carácter de s_L, buscar si s_C empieza aquí. Para eso, carácter por carácter de s_C, verificar si el carácter de s_C corresponde al de s_L. Si no, seguir en el ciclo exterior. Cada vez que tenemos todos los caracteres de s_C encontrados, incrementamos un contador. Al final, regresamos este contador.

Problema 5 [1.5 punto]

De por hecho la siguiente información y solamente implemente el programa:

Se puede demostrar que ciertas matrices cuadradas $\mathbf{A}$ de dimensiones $n \times n$ se pueden factorizar de la manera

$$\mathbf{A} = \mathbf{L}\mathbf{L}^T,$$

donde $\mathbf{L}$ es una matriz triangular inferior (es decir, que sólo tiene valores diferentes de cero en la diagonal principal y abajo de ella).

Las fórmulas para calcular las entradas de $\mathbf{L}$ para la columna $j = 1, \ldots, n$, y el renglón $i = j+1, \ldots, n$, son:

- $l_{jj}^2 = a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2$ para los elementos en la diagonal principal de $\mathbf{L}$, y
- $l_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk}}{l_{jj}}$ para el resto de los elementos,

donde l_{ij} son los elementos de la matriz $\mathbf{L}$.

Escriba un programa que dada una matriz $\mathbf{A}$ (suponga que cumple las condiciones), encuentra la matriz $\mathbf{L}$.

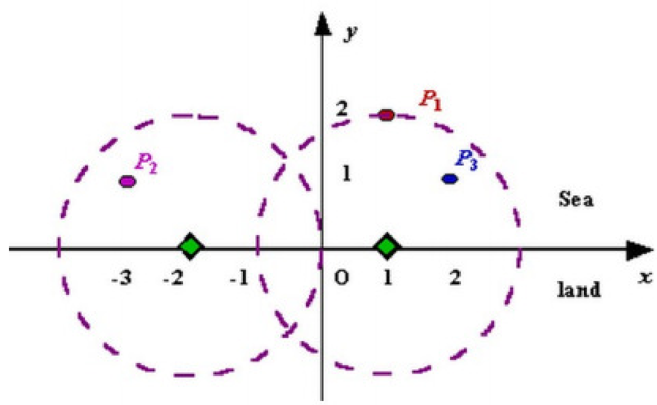
Solution: Programar 1 ciclo que va recorriendo de manera creciente las columnas de $\mathbf{L}$. Anidado al anterior, otro ciclo que recorre de la diagonal hacia abajo los renglones de $\mathbf{L}$, los dos ciclos anteriores se colocan en cada entrada de $\mathbf{L}$. Anidado a los otros 2 ciclos, para cada entrada programar un ciclo que usa la 1era o la 2da fórmula según sea el caso.

Problema 6 [1.5 punto]

Supongamos que la costa es una línea recta infinita, representada como el eje x en la figura abajo. En la zona del mar hay un conjunto de N islas, representadas por P_1, P_2 y P_3 en la figura (en este caso $N = 3$). El objetivo de este problema es colocar un conjunto mínimo de radares (rombos verdes en la figura) sobre la costa (es decir, el eje x) de forma que todas las islas queden cubiertas por alguno de los radares. Para ello, considere que el radio de acción de cada radar es r.

- **(0.5 puntos)** Dada la posición de una isla (x, y), y el radio de acción de los radares (r), desarrolle una función que calcule y devuelva el rango de valores de x en que podemos colocar un radar de forma que cubra a dicha isla. Considere que el radar se va a poner justo en la línea de costa.
- **(0.5 puntos)** Supongamos que le han indicado cual es la isla P_{min} de coordenada x mínima (la más a la izquierda en la figura). Escriba una función que determine dónde colocar un radar de tal manera que, independientemente de qué otras islas hay, el radar: 1) cubra P_{min} y 2) cubra posiblemente un máximo de otras islas.
- **(0.5 puntos)** Usando las respuestas a las preguntas anteriores, desarrolle una función que dado un conjunto de N islas ($N \leq 100,000$), y el radio r, determine el número mínimo de radares necesarios para cubrir todas las islas. Si el problema no tiene solución, es decir, no se pueden cubrir todas las islas, retorne -1. Describa el método, razone en palabras por qué funciona e incluya un pseudocódigo para resolver este problema.

  Ejemplo: Con los datos de entrada representados en la figura, su función debería devolver el número 2, ya que situando los radares en los rombos verdes quedan cubiertas todas las islas, mientras que con poner solamente un radar no se pueden cubrir las 3 islas.



Solution:

- Los dos casos extremos corresponden a situaciones en las cuales las islas están justo sobre el círculo de acción. Se trata entonces de resolver:

$$(x - x_i)^2 + y_i^2 + r^2,$$

  y la dos soluciones darán el intervalo en el cual se puede colocar el radar.

- Usar un algoritmo glotón: calcular primero para todas las islas su intervalo de ubicación del radar; ordenarlas por valor del mínimo del intervalo. Procesar esos valores: Para cada isla, en el orden determinado arriba, colocar el radar al máximo del intervalo. Para las islas siguientes, ver si se pueden colocar bajo el primer radar, sino aplicar el mismo procedimiento (es decir, colocar un radar lo más a la derecha posible).