



Evaluaciones- CP - Todas las prácticas

Complejidad Algorítmica (Universidad Peruana de Ciencias Aplicadas)



Escanea para abrir en Studocu

PC1:

Pregunta 1

0 de 4 puntos

Escriba un algoritmo en Python que implemente una **función recursiva** llamada `tot_palabras_archivo`, que reciba por parametro un archivo de texto (`lorem.txt`) conteniendo `n` lineas (ver adjunto [lorem.txt](#)), y retorne el numero de palabras contenidos en este.

Ejemplo:

`file=lorem.txt`

Input

`tot_palabras_archivo(file)`

Output

5000

Descargar archivo de ejemplo:

[lorem.txt](#)

Pregunta 2

4 de 4 puntos

Escriba un algoritmo en Python que implemente una **función recursiva** que reciba como parámetro una lista conteniendo ya sea ítems de números enteros y sublistas de enteros, y devuelva la suma total de todos los ítems.

Ejemplo:

Input

`lista = [1, 2, [3,4], [5,6]]`

`SumaEnterosdeLista (lista)`

Output

21

(equivale a $(1+2) + (3+4) + (5+6)$)

Pregunta 3

4 de 4 puntos

Escriba un algoritmo en Python que implemente una **función recursiva** que reciba como parámetro un número `n`, ($n > 0$), y calcule la suma de los enteros positivos de $n+(n-2) + (n-4) \dots$ (hasta que $n-x \leq 0$).

Input

`n = 6`

`Suma_Series (n)`

Output

12

(equivale a $6 + (6-2) + (6-4) - (6-6)$)

Pregunta 4

2 de 2 puntos

Teniendo el siguiente código ejemplo, calcule la ecuación polinomial que resulta de contar los pasos u operaciones que en ella se realizan y seleccione la respuesta correcta:

```
def f(x):
    respuesta = 0
    for i in range(x):
        respuesta += x
    for i in range(10):
        respuesta += 1

    for i in range(x):
        for j in range(x):
            respuesta += 1

    return respuesta
```

$$- + (+ x + 10) \\ + x^2 + x \\ 11 + x^2 + 2x$$

- Respuestas:
1. $12 + x + x^2$
 2. $12 + 2x + x^2$ ☒
 3. $10 + x + x^2$
 4. Ninguna es una respuesta correcta.
 5. $11 + x + x^2$

Pregunta 5

2 de 2 puntos

El la opción que más se ajusta a la definición de ESPACIO DE BUSQUEDA para la resolución de problemas.

- Respuestas:
1. El espacio de búsqueda es el total de soluciones existentes a un problema.
 2. El espacio de búsqueda lo conforman todas las combinaciones existentes para solucionar un problema, sean correctas o no.
 3. El espacio de búsqueda son todas las combinaciones correctas que dan solución a un problema.
 4. El espacio de búsqueda es la representación espacial de un grafo, en donde se buscarán las aristas.

Pregunta 6

2 de 2 puntos

Relacione de forma correcta la notación Big O con la instrucción y/o estructura de código respectivo:

Pregunta

$O(\log n)$: Complejidad Logarítmica. **c**

$O(1)$: Complejidad Constante. **d**

$O(n)$: Complejidad Linear **a**

$O(n^2)$ Complejidad cuadrática **b**

Todas las opciones de respuesta

- a. Recorrer un bucle for, foreach, while, do while
- b. Bucle anidado
- c. Funciones recursivas.
- d. print(matriz[0])

Pregunta 7

2 de 2 puntos

La técnica de Fuerza Bruta utiliza combinaciones y permutaciones. Elija la afirmación correcta:

- Respuestas:
1. La ordenación por selección (Selection Sort) es un ejemplo de algoritmo que aplica FB utilizando permutaciones. ✓
 2. Una permutación es una combinación ordenada. ✓
 3. La técnica de FB utiliza **permutaciones** (con o sin repetición) si el orden si importa. ✓
 4. La técnica de FB utiliza **combinaciones** (con o sin repetición) si el orden no importa. ✓
 5. Todas las afirmaciones son correctas. ✓

domingo 26 de noviembre de 2023 18H03' PET

← Aceptar

Evaluación Parcial:

Pregunta 1

2 de 2 puntos

¿Cuánto espacio es necesario para almacenar una lista de adyacencia, en función a n , el número de vértices y m , el número de aristas?

Elija la respuesta correcta.

Respuesta seleccionada:

☒ c. $O(m+n)$

Respuestas:

a. $O(m^2)$

b. $O(n)$

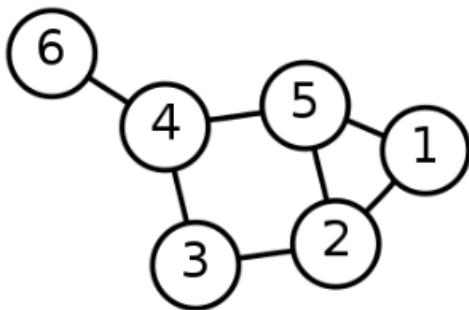
☒ c. $O(m+n)$

d. $O(m)$

Pregunta 2

2 de 2 puntos

Dado el siguiente Grafo, indique el grado del vértice 1:

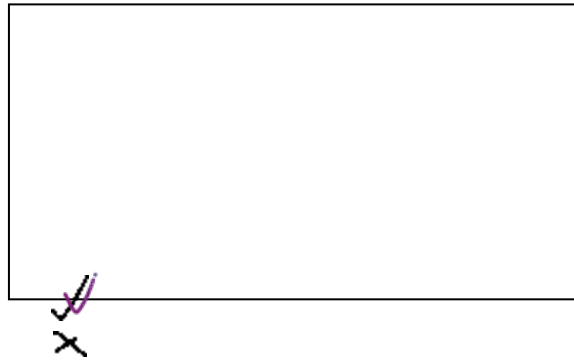


Nota: su respuesta debe ser ingresada como un valor entero

Respuesta seleccionada: ☒ 2

Respuesta correcta: ☒ 2

Rango de respuesta +/- 0 (2 - 2)



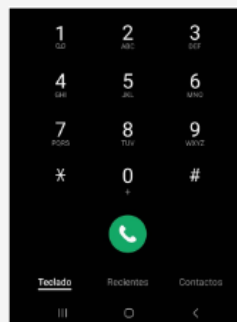
Pregunta 5

5 de 6 puntos

Todos conocemos y usamos el teclado de nuestros smartphones, y también sabemos que los dígitos permiten representar letras.

Por ejemplo, el dígito 2 representa a las letras A, B y C, el dígito 3 representa a las letras D, E y F, etc. Pero solo se consideran los dígitos del 2 al 9 (inclusive), dado que los dígitos 0 y 1 no tienen ninguna letra asignada.

Como se observa en la imagen:



Por ejemplo,

- Si se presiona el 2 y el 4, los resultados posibles serían: ["AG", "AH", "AI", "BG", "BH", "BI", "CG", "CH", "CI"]
- Si se presiona el 0 y el 5, el resultado sería ["NO"]
- Si se presiona el 1 y el 1, el resultado sería ["NO"]

Como sabemos que Ud. es un excelente programador en lenguaje Python se le pide generar el resultado anterior a partir de la lectura de dos teclas numéricas que deberán ser obtenidas como **datos de entrada desde el archivo de texto** (historial_marcacion.txt) que representa el historial de teclas presionadas:

Su programa deberá leer dicho archivo y procesar cada línea, según la siguiente estructura:

Ejemplo de entrada (archivo: historial_marcacion.txt)

```
2 4
0 5
5 8
4 1
1 1
3 7
0 0
```

Donde cada línea representa las teclas numéricas presionadas en cada intento.

Ejemplo de salida

```
["AG", "AH", "AI", "BG", "BH", "BI", "CG", "CH", "CI"]
["NO"]
["JT", "JU", "JV", "KT", "KU", "KV", "LT", "LU", "LV"]
["NO"]
["NO"]
["DP", "DQ", "DR", "DS", "EP", "EQ", "ER", "ES", "FP", "FQ", "FR", "FS"]
["NO"]
```

IMPORTANTE: si el archivo se encuentra vacío, la salida esperada será un mensaje: "Ninguna tecla presionada".

Descargue el archivo: [historial_marcacion.txt](#)

Se solicita:

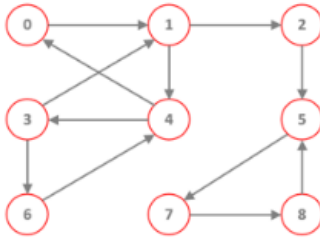
- Desarrollar un código en Python que lea la entrada de datos (según el contenido de archivo historial_marcacion.txt) y produzca la salida deseada.
- Indicar que algoritmo tomó de base para solucionar este ejercicio y por qué.
- Adjunte su archivo respuesta en formato *.ipynb o *.py

Pregunta 6

4 de 6 puntos

El siguiente grafo representa a un grupo de 9 personas que pertenecen a una red social.

Cada persona es representada por un Nodo o Vértice en el Grafo, la etiqueta del nodo es el identificador de la persona en la red social. El grafo nos brinda información acerca de cómo una persona es seguidora (follower) de otra persona en la red social. Así, por ejemplo, podemos interpretar que la Persona_0 sigue a la Persona_1 y la Persona_8 es seguida por la Persona_7.



Se solicita desarrolle un código en lenguaje Python que permita:

- (5 puntos) Identificar los grupos de personas que estén estrechamente vinculados debido a sus múltiples relaciones, gustos, lugares comunes y preferencias en la red social; de tal forma que luego se pueda usar este algoritmo para en un sistema que pueda gestionar recomendaciones para quienes conforman estos componentes.
 - La salida o resultado debe considerar :
 - Visualizar el grafo dirigido original.
 - Mostrar el resultado de la variable conteniendo los grupos de personas.
 - Visualizar el grafo dirigido resultante.
- (1 punto) Indicar el algoritmo o técnica utilizado para resolver el requerimiento.

Adjunte su archivo respuesta en formato *.ipynb o *.py. No olvidar adjuntar el archivo de tipo imagen correspondiente a la visualización (de ser el caso).

Respuesta seleccionada: 2 - pregunta ep.py

domingo 26 de noviembre de 2023 18H17' PET

← Aceptar

PC2:

Pregunta 1

2 de 2 puntos

Sobre Union-Find Disjoint Sets (UFDS), seleccione la afirmación incorrecta:

Respuesta seleccionada: ☒ D. Permite la detección de los componentes fuertemente conectados.

- Respuestas:
- ☐ A. Un conjunto disjunto es un grupo de conjuntos donde ningún elemento puede estar en más de un conjunto.
 - ☐ B. Permite detectar ciclos dentro de un grafo.
 - ☐ C. Es una estructura de datos que almacena una colección de conjuntos disjuntos (no superpuestos).
 - ☒ D. Permite la detección de los componentes fuertemente conectados.

Pregunta 2

2 de 2 puntos

El algoritmo de búsqueda rápida Quick-Find intenta mejorar al algoritmo Union-Find para que sea más eficiente.

Respuesta seleccionada: ☒ Falso

- Respuestas:
- ☐ Verdadero
 - ☒ Falso

Pregunta 3

2 de 2 puntos

Marque una o más respuestas que considere correctas.

Respuestas seleccionadas: ☒ A. Kruskal y PRIM son algoritmos de enfoque voraz o codicioso.

- Respuestas:
- ☒ A. Kruskal y PRIM son algoritmos de enfoque voraz o codicioso.
 - ☒ B. PRIM verifica y selecciona los vértices de menor costo antes de consultar por la formación de ciclo.
 - ☒ C. PRIM ordena las aristas del grafo G, de menor a mayor antes de verificar si al considerarla conformara un ciclo.
 - ☒ D. Todas las afirmaciones son correctas.
 - ☒ E. Kruskal y PRIM consultan por la formación de ciclos.

Pregunta 4

2 de 2 puntos

Marque la(s) respuesta(s) que considere incorrecta(s):

- Respuestas seleccionadas: ☒ C.
Un MST es un subconjunto de los bordes ponderados de un grafo dirigido que conecta todos los vértices entre sí y que cumple con todas las características de ser un grafo de tipo árbol.
- Respuestas: A. Un MST es un grafo que se extiende a partir de un grafo no dirigido y conexo donde el peso total de los bordes sea el mínimo posible.
B. Un MST constituye un modelo de optimización de redes.
☒ C.
Un MST es un subconjunto de los bordes ponderados de un grafo dirigido que conecta todos los vértices entre sí y que cumple con todas las características de ser un grafo de tipo árbol.
D. Un MST es llamado también árbol de expansión de peso mínimo.

Pregunta 5

2 de 2 puntos

Señale si es verdadera o falsa la siguiente afirmación:

Así como se puede obtener varios árboles de expansión mínima a partir de un grafo no dirigido ponderado, es factible también encontrar más un árbol de expansión con el mismo peso mínimo (peso total de los bordes sea el mínimo posible).

- Respuesta seleccionada: ☒ Verdadero
- Respuestas: ☒ Verdadero
☐ Falso

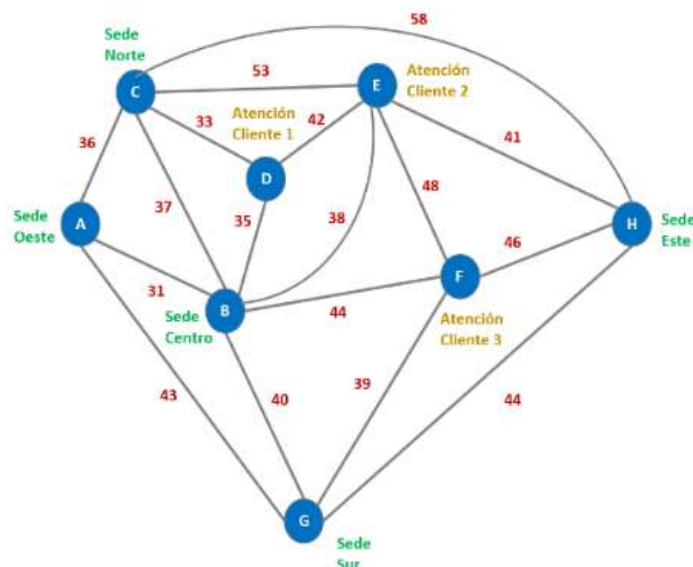
Pregunta 6

6 de 6 puntos

Una empresa de telecomunicaciones acaba de incursionar en una nueva ciudad y está dimensionando y planificando el despliegue de un sistema para interconectar sus servidores prime.

La interconexión debe unir los servidores prime que se encuentran en los centros de datos de sus cinco sedes (norte, sur, centro, este y oeste), así como también interconectar a sus tres centros de atención al cliente.

El área de Planificación y Optimización debe seleccionar un conjunto de líneas de fibra óptica que conecte todos los servidores prime de sus 5 sedes y de sus 3 centros de atención al cliente, a un mínimo costo posible por línea.



Se solicita:

- (1pto) Hallar de forma manual, si existe, el conjunto de líneas de fibra óptica.
- (1ptos) Hallar de forma manual el costo total del conjunto de líneas elegido, adjuntando el procedimiento seguido de forma manual en un archivo tipo .jpg
- (4ptos) Indicar el algoritmo o técnica utilizado para resolver el requerimiento, implementando y adjuntando el código en Python que considere:
 - Ingresar y mostrar el grafo del enunciado.
 - Señalar cada una de las líneas de fibra óptica encontradas así como sus costos por cada línea (camino) de fibra óptica.
 - Mostrar el grafo resultante con las líneas de fibra óptica encontradas así como sus costos por cada línea (camino) de fibra óptica.

Respuesta seleccionada: ejercicio6-pc.zip

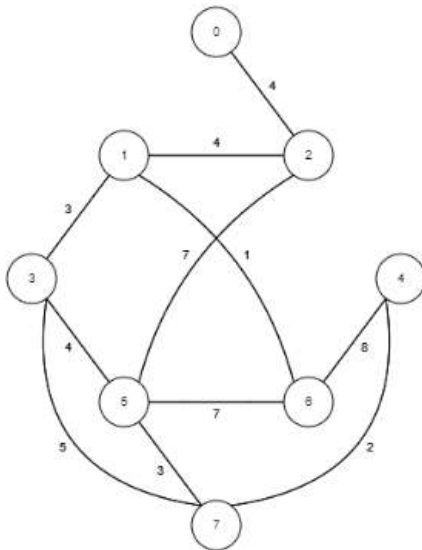
Comentarios para respuesta: [No se ha dado ninguna]

Pregunta 7

4 de 4 puntos

A partir del siguiente grafo ponderado no dirigido, aplique los pasos del Algoritmo PRIM para:

- (1pto) De forma manual: Construir el árbol de expansión mínimo (MST) a partir del nodo 7
- (1pto) De forma manual: Determinar el peso del MST obtenido.
- (2 pts) Utilizando el código del Algoritmo PRIM, refrendar la solución manual a nivel de grafo y peso mínimo del MST resultante en a) y b)



Respuesta seleccionada: [ejercicio7-pc.zip](#)

Comentarios para respuesta: [No se ha dado ninguna]

domingo 26 de noviembre de 2023 18H21' PET

← Aceptar

EF:

Pregunta 1

0 de 2 puntos



Si aplicamos un criterio de selección voraz a un problema de decisión, entonces verificamos que (señale una respuesta correcta):

Respuesta seleccionada:

☒ d. Al menos una solución factible está garantizada.

Respuestas:

☐ a. La solución óptima está garantizada.

☐ b.

El cálculo de la solución óptima se obtiene al tomar una decisión en cada paso considerando el problema actual y la solución al problema resuelto previamente.

☒ c. Ninguna de las otras opciones es cierta.

☐ d. Al menos una solución factible está garantizada.

Pregunta 2

2 de 2 puntos



Podemos aplicar un algoritmo voraz o codicioso en la solución de un problema cuando se cumplen las siguientes condiciones (marque las afirmaciones que crea correctas).

Respuestas
seleccionadas:

☒ b.
Si se puede encontrar una solución óptima al problema eligiendo la mejor opción en cada paso sin reconsiderar los pasos anteriores una vez elegidos.

Respuestas:

☒ d. Si la solución general óptima del problema corresponde a la solución óptima de sus subproblemas.
a. Si la solución general óptima del problema está garantizada.

☒ b.
Si se puede encontrar una solución óptima al problema eligiendo la mejor opción en cada paso sin reconsiderar los pasos anteriores una vez elegidos.

c. Todas las afirmaciones son correctas.

☒ d. Si la solución general óptima del problema corresponde a la solución óptima de sus subproblemas.

Pregunta 3

0,66666 de 2 puntos



Kruskal y PRIM son los algoritmos que resuelven el cálculo del árbol de expansión mínimo (MST). Son características que comparten ambos algoritmos (seleccione todas las opciones que crea correctas).

Respuestas
seleccionadas:

☒ b. Giran en torno a verificar si al agregar un borde (PRIM) o un vértice (KRUSKAL) se crea un ciclo o no.

☒ c. Son algoritmos de tipo voraz.

Respuestas:

☒ a. Kruskal comprueba si una arista (v, w) forma un ciclo si v y w están en el mismo componente conexo.

b. Giran en torno a verificar si al agregar un borde (PRIM) o un vértice (KRUSKAL) se crea un ciclo o no.

☒ c. Son algoritmos de tipo voraz.

☒ d.

En cada iteración PRIM marca un nuevo vértice que sea contiguo al seleccionado(s) previamente, pero seleccionando la arista con el peso más bajo sin que forme un ciclo.



Los Linterna Verde se encuentran abrumados por la cantidad de delitos a lo largo de todo el universo, por lo que para poder organizar mejor los esfuerzos para proporcionar ayuda de la manera más eficiente, han decidido utilizar expertos en Complejidad Algorítmica del sector 2814 ubicados en la UPC, Lima, Peru.

El plan

- Formar una cantidad determinada de equipos especializados para ser enviados a regiones con problemas relacionados entre sí.
- Para llevar a cabo el plan, se tienen registros de las zonas más peligrosas y su ubicación en el espacio.
- Por favor ayude a los Linterna Verde a elegir los sectores donde enviar a cada equipo.
- Los equipos han sido organizados en base a su experiencia y poder de combate, de modo que el primer equipo será enviado al sector más numeroso, el siguiente equipo al siguiente sector, etc.
- Los sectores deben ser definidos por grupos de zonas en base a la distancia euclidiana entre ellas.

Considere los siguientes datos de entrada:

```
2
5
1 1 1
2 2 2
1 2 1
4 4 4
4 4 3
2
3
1 2 3
3 2 1
100 200 300
2
```

Donde:

- La primera línea inicia con un número T , que representa el número de casos de prueba.
- Cada caso de prueba inicia con N , el número de zonas peligrosas.
- Luego siguen N líneas, cada una conteniendo tres números, X , Y y Z , que representan las coordenadas de cada zona.
- Finalmente sigue un número K , que determina el número de equipos de linternas verdes formados.

Se solicita:

- (0.5 pts.) Adquirir los datos desde este archivo: [linternav.txt](#) (no considerar ningún ingreso de datos por consola o de forma manual).
- (2.0 pts.) Construir el grafo a partir de ejecutar a).
- (0.5 pts.) Identificar el algoritmo que resuelve el problema.
- (3.5 pts.) Implementar correctamente aplicando el algoritmo identificado en c).
- (0.5 pts.) Salida: Por cada caso de prueba, calcular y mostrar el número de zonas por cada sector

Ejemplo de salida para los datos brindados

```
....
3
2
2
1
....
```

Donde:

Por cada caso de prueba, se debe mostrar una línea por cada equipo de linternas en dicho caso de prueba conteniendo el número de zonas por cada sector, empezando por el mayor y terminando con el menor. Un espacio en blanco entre cada caso de prueba es necesario para poder diferenciarlos.



¡La carrera decide!



Alejandra, una estudiante de informática, es muy aficionada a las carreras a campo traviesa y participa en las carreras que se organizan todos los domingos en un gran parque de su ciudad.

Cada uno de los participantes en la carrera obtiene una tarjeta de ruta, que especifica una secuencia de puntos de control, que deben visitar en el orden indicado.

Alejandra es una chica muy atractiva y varios corredores le han pedido una cita. A ella le gustaría elegir a uno de ellos durante la carrera, por ello invitó el domingo a todos sus admiradores al parque y dejó que la carrera decidiera:

¡El ganador sería el que obtenga el máximo número de puntos!

Alejandra ideó las siguientes reglas:

- Un corredor anota un punto si se encuentra con ella en el punto de control.
- Si un corredor anotó un punto en el punto de control, entonces no puede obtener otro punto a menos que él y ella se muevan a los siguientes puntos de control especificados en sus tarjetas.
- La ruta especificada por la tarjeta puede cruzar el mismo punto de control más de una vez, cada competidor debe seguir estrictamente las instrucciones de carrera escritas en su tarjeta.
- Entre dos encuentros consecutivos, Alejandra y los competidores pueden visitar cualquier número de puntos de control.
- Los chicos realmente harán lo mejor que puedan, por lo que puede suponer que cada uno de ellos podrá visitar cualquier cantidad de puntos de control mientras Alejandra corre entre dos puntos consecutivos en su ruta.

INPUT

- Hay un entero d en la primera línea del archivo de entrada, $1 \leq d \leq 10$. Este es el número de conjuntos de datos a procesar.
- A partir de la segunda línea, se muestra el detalle de cada conjunto de datos (hasta d), donde cada conjunto de datos consta de varias líneas, y la primera especifica la ruta que figura en la tarjeta de carrera de Alejandra. Las líneas consecutivas contienen las rutas presentadas en la tarjeta a Diego.
- Se le presenta al menos una ruta a Diego. Esta ruta se proporciona como una secuencia de números enteros del intervalo $[1, 1000]$ separados por espacios simples.
- El número 0 representa el final de la ruta, aunque cuando se coloca al principio de la línea significa el final del conjunto de datos.
- Hay al menos dos y como máximo 2000 puntos de control en una tarjeta de carrera.

OUTPUT

- La i -ésima línea del archivo de salida debe contener un número entero.
- Ese número entero debe ser igual a la mayor cantidad de veces que Diego puede reunirse con Alejandra para las rutas presentadas en la tarjetas de carreras dadas a Diego en el i -ésimo conjunto de datos.

Ejemplo

Ejemplo de entrada:

```
3
1 2 3 4 5 6 7 8 9 0
1 3 8 2 0
2 5 7 8 9 0
1 1 1 1 1 2 3 0
1 3 1 3 5 7 8 9 3 4 0
1 2 3 5 0
0
1 3 5 7 0
3 7 5 1 0
0
1 2 1 1 0
1 1 1 0
0
```

Salida de muestra:

```
6
2
3
```

Se solicita:

(a) 6ptos Escribir un algoritmo en Python que implemente el INPUT/OUTPUT anteriormente descrito, considerando:

- Leer el contenido de la tarjeta de carrera de Alejandra y el contenido de las rutas presentadas en la tarjeta a Diego que se detallan en el siguiente archivo: [carrera.txt](#) (no considerar ningún ingreso de datos por consola o de forma manual).
- Calcular el mayor número de veces que Diego puede encontrarse con Alejandra durante la carrera

(b) 1 pto Sustentar que técnica y/o tipo de algoritmo ha utilizado para implementar la solución.