Informe proyecto final

Integrantes:

202259570 Maria Paula Lemus Agudelo
202259594 Jandi Yandubery Ramírez
202259461 Camilo Andrés Viedma Rueda

Universidad del valle

Tuluá

Orientador:

Carlos delgado

Área:

Análisis de al de algoritmos

- Análisis temporal y espacial de las operaciones implementadas

  Se adjuntan las tablas de los análisis temporales y espaciales implementados

  -Tiempo:

VALOR ENTRADA	1	2	3	4	5	6	7	8	9	10
4	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
8	0	0	0,00150824	0	0	0	0	0	0	0
10	0,00100136	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0,00099134	0	0	0	0
16	0	0	0	0,00351238	0	0,00101042	0	0,00099611	0,0010047	0
18	0	0	0	0	0,00099826	0	0,00078797	0,00099874	0	0
20	0	0	0,00102448	0	0	0,00100207	0	0,00102592	0	0

VALOR ENTRADA	PROMEDIO	1	2	3	4	5	6	7	8	9	10	SUMA	1/N	DESVIACION
4	0,000000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0	0	0
6	0,000000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0	0	0
8	1,508240E-04	2,2748E-08	2,2748E-08	1,8426E-06	2,2748E-08	2,2748E-08	2,2748E-08	2,2748E-08	2,2748E-08	2,2748E-08	2,2748E-08	2,047E-06	2,2748E-07	0,00047695
10	1,001360E-04	8,1220E-07	1,0027E-08	1,0027E-08	1,0027E-08	1,0027E-08	1,0027E-08	1,0027E-08	1,0027E-08	1,0027E-08	1,0027E-08	9,024E-07	1,0027E-07	0,00031666
12	0,000000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0	0	0
14	9,913400E-05	9,8275E-09	9,8275E-09	9,8275E-09	9,8275E-09	9,8275E-09	7,9603E-07	9,8275E-09	9,8275E-09	9,8275E-09	9,8275E-09	8,845E-07	9,8275E-08	0,00031349
16	6,523610E-04	4,2557E-07	4,2557E-07	4,2557E-07	8,1797E-06	4,2557E-07	1,2821E-07	4,2557E-07	1,1816E-07	1,2414E-07	4,2557E-07	1,11E-05	1,2337E-06	0,00111074
18	2,784970E-04	7,7561E-08	7,7561E-08	7,7561E-08	7,7561E-08	5,1806E-07	7,7561E-08	2,5956E-07	5,1875E-07	7,7561E-08	7,7561E-08	1,839E-06	2,0437E-07	0,00045207
20	3,052470E-04	9,3176E-08	9,3176E-08	5,1730E-07	9,3176E-08	9,3176E-08	4,8556E-07	9,3176E-08	5,1937E-07	9,3176E-08	9,3176E-08	2,174E-06	2,4161E-07	0,00049153

- espacio

VALOR ENTRADA	1	2	3	4	5	6	7	8	9	10
4	8	0	0	0	0	0	0	0	0	8
6	0	0	0	0	0	4	4	4	4	4
8	0	0	0	0	0	0	0	0	0	0
10	4	0	0	0	4	4	0	0	0	0
12	4	0	0	0	0	0	4	4	4	0
14	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0
18	8	0	0	0	0	0	0	0	0	0
20	16	0	0	0	0	0	0	0	0	0

VALOR ENTRADA	PROMEDIO	1	2	3	4	5	6	7	8	9	10	SUMA	1/N	DESVIACION
4	1,600000E+00	4,0960E+01	2,5600E+00	2,5600E+00	2,5600E+00	2,5600E+00	2,5600E+00	2,5600E+00	2,5600E+00	2,5600E+00	4,0960E+01	102,4	11,3777778	3,37309617
6	2,000000E+00	4,0000E+00	4,0000E+00	4,0000E+00	4,0000E+00	4,0000E+00	4,0000E+00	4,0000E+00	4,0000E+00	4,0000E+00	4,0000E+00	40	4,44444444	2,10818511
8	0,000000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0	0	0
10	1,200000E+00	1,6000E+01	0,0000E+00	0,0000E+00	0,0000E+00	1,6000E+01	1,6000E+01	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	48	5,33333333	2,30940108
12	1,600000E+00	5,7600E+00	2,5600E+00	2,5600E+00	2,5600E+00	2,5600E+00	2,5600E+00	5,7600E+00	5,7600E+00	5,7600E+00	2,5600E+00	38,4	4,26666667	2,06559112
14	0,000000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0	0	0
16	0,000000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0,0000E+00	0	0	0
18	8,000000E-01	5,1840E+01	6,4000E-01	6,4000E-01	6,4000E-01	6,4000E-01	6,4000E-01	6,4000E-01	6,4000E-01	6,4000E-01	6,4000E-01	57,6	6,4	2,52982213
20	1,600000E+00	2,0736E+02	2,5600E+00	2,5600E+00	2,5600E+00	2,5600E+00	2,5600E+00	2,5600E+00	2,5600E+00	2,5600E+00	2,5600E+00	230,4	25,6	5,05964426

b) Análisis de cercanía al optimo

Su cercanía a lo óptimo es buena, aunque su complejidad será elevada mientras el número de entradas aumente así que siempre dependerá de la cantidad de datos ingresada

c) Detalles principales de la implementación (no el código).

- **Calendario:**

Este script en Python utiliza la biblioteca Tkinter para crear una interfaz gráfica que permite al usuario seleccionar un archivo de texto con datos específicos y generar un calendario de partidos de fútbol. A continuación, se presentan los detalles principales de la implementación:

1. Interfaz Gráfica (Tkinter):

- Se utiliza la biblioteca Tkinter para crear la interfaz gráfica.

- Se crea una ventana principal (`root`) con el título "Generador y Verificador de Calendario".

- Se agrega un botón ("Cargar Archivo") que, al hacer clic, ejecuta la función `elegir_archivo`.

- Se utiliza un widget de texto (`output_text`) para mostrar el resultado y el calendario generado.

2. Funciones Principales:

- `elegir_archivo`:

- Utiliza `filedialog.askopenfilename` para permitir al usuario seleccionar un archivo de texto.

- Llama a la función `leer_datos_desde_archivo` para obtener los datos del archivo seleccionado.

- Llama a la función `generar_calendario` con los datos obtenidos y muestra el resultado en el widget de texto.

- `leer_datos_desde_archivo(ruta)`:

- Lee los datos de un archivo de texto, incluyendo el número de equipos, límites de partidos y una matriz de distancias.

- Devuelve los datos como una tupla (n_equipos, min_partidos, max_partidos, distancias).

- `generar_calendario (n_equipos, min_partidos, max_partidos, distancias) `:

- Genera un calendario de partidos para un torneo de equipos utilizando el algoritmo de emparejamiento.

- Asegura que el número de equipos sea par.

- Devuelve el calendario como una lista de listas.

3. Otras Funciones:

- **`generar_calendario_desde_archivo(ruta)`:

- Lee los datos desde un archivo y genera el calendario utilizando las funciones anteriores.

- `imprimir_calendario (calendario, n_equipos, min_partidos, max_partidos)`:

- Imprime el calendario en la consola con información sobre el número de equipos y límites de partidos.

4. **Ventana Principal:

- Se crea una instancia de `Tk ()` para la ventana principal.

- Se agrega un botón para cargar el archivo y un widget de texto para mostrar el resultado.

5. Línea de Ejecución:

- El bloque `if __name__ == "__main__":` inicia la interfaz gráfica al ejecutar el script.

En resumen, el script proporciona una herramienta gráfica para generar calendarios de partidos de fútbol a partir de datos de un archivo, ofreciendo una interfaz amigable para los usuarios.

Sol ingenua:

Este script de Python utiliza la biblioteca tkinter para crear una interfaz gráfica simple que permite cargar un archivo de entrada, procesar los datos y generar un calendario. Aquí están los detalles principales de la implementación:

1. **Interfaz Gráfica con Tkinter:

   - Se importa la biblioteca `tkinter` para crear la interfaz gráfica.

   - Se define una función `seleccionar_archivos` que se llama cuando se hace clic en el botón "Cargar Archivo".

   - Se utiliza `filedialog.askopenfilename` para abrir un cuadro de diálogo de selección de archivo y obtener la ruta del archivo seleccionado.

2. Procesamiento de Archivo de Entrada:

   - La función `leer_datos_desde_archivo` lee el archivo seleccionado y extrae información importante, como el número de equipos, el mínimo y máximo de partidos, y las distancias entre los equipos.

   - Los datos se devuelven en una tupla (num_equipos, min_partidos, max_partidos, distancias).

3. Generación de Calendario:

   - La función `generar_calendario` crea un calendario de partidos basado en los datos proporcionados.

   - Utiliza un algoritmo para asignar partidos de manera que se minimice la distancia entre los equipos, evitando repeticiones y cumpliendo con las restricciones de mínimo y máximo de partidos.

   - El calendario generado se muestra en el widget `output_text` de la interfaz gráfica.

4. **Interfaz Gráfica - Tkinter:

   - Se crea una ventana principal con `tk. Tk()` y se le asigna un título.

- Se agrega un botón ("Cargar Archivo") que, al hacer clic, llama a la función `seleccionar_archivos`.

- Se incluye un widget `Text` llamado `output_text` para mostrar el resultado y los detalles del calendario generado.

5. Ejecución del Programa:

- La ejecución principal (`if __name__ == "__main__":`) inicia la interfaz gráfica mediante `root.mainloop()`.

6. **Manejo de Errores: **

- Se utiliza un bloque `try-except` al leer el archivo para manejar posibles errores, y se imprime un mensaje en caso de error.

- Si se produce un error al leer el archivo, se muestra un mensaje de error en el widget `output_text`.

En resumen, este script utiliza Tkinter para proporcionar una interfaz gráfica que facilita la carga de archivos de entrada, el procesamiento de datos y la visualización del calendario generado.

d) Descripción y análisis de las pruebas realizadas

a) Pruebas realizadas:

Cantidad de pruebas por cantidad de equipos: 1000

Tiempo por Algoritmo calendario

Número de equipos: 4, Tiempo promedio de ejecución: 0.00000000 segundos

Número de equipos: 6, Tiempo promedio de ejecución: 0.00001251 segundos

Número de equipos: 8, Tiempo promedio de ejecución: 0.00004675 segundos

Número de equipos: 10, Tiempo promedio de ejecución: 0.00007848 segundos

Número de equipos: 12, Tiempo promedio de ejecución: 0.00010343 segundos

Número de equipos: 14, Tiempo promedio de ejecución: 0.00013318 segundos

Número de equipos: 16, Tiempo promedio de ejecución: 0.00017643 segundos

Número de equipos: 18, Tiempo promedio de ejecución: 0.00025149 segundos

Número de equipos: 20, Tiempo promedio de ejecución: 0.00033384 segundos

Número de equipos: 30, Tiempo promedio de ejecución: 0.00098407 segundos

Número de equipos: 36, Tiempo promedio de ejecución: 0.00150464 segundos

Cantidad de pruebas por cantidad de equipos: 1000

Tiempo por Solución ingenua

Número de equipos: 4, Tiempo promedio de ejecución: 0.00002324 segundos

Número de equipos: 6, Tiempo promedio de ejecución: 0.00003245 segundos

Número de equipos: 8, Tiempo promedio de ejecución: 0.00005685 segundos

Número de equipos: 10, Tiempo promedio de ejecución: 0.00013499 segundos

Número de equipos: 12, Tiempo promedio de ejecución: 0.00021015 segundos

Número de equipos: 14, Tiempo promedio de ejecución: 0.00030010 segundos

Número de equipos: 16, Tiempo promedio de ejecución: 0.00039095 segundos

Número de equipos: 18, Tiempo promedio de ejecución: 0.00055608 segundos

Número de equipos: 20, Tiempo promedio de ejecución: 0.00071611 segundos

Número de equipos: 30, Tiempo promedio de ejecución: 0.00217678 segundos

Número de equipos: 36, Tiempo promedio de ejecución: 0.00362082 segundos

Análisis:
Esta prueba es un estudio comparativo de dos algoritmos de generación de calendarios para diferentes cantidades de equipos. Los dos algoritmos evaluados son el "calendario" y la "solución ingenua". A continuación, se describen algunos aspectos clave de esta prueba:

1. Cantidad de Pruebas:

   - Se realizaron 1000 pruebas para cada cantidad de equipos evaluada.

2. Algoritmos Evaluados:

   - Se comparan dos algoritmos: "calendario" y "solución ingenua".

3. Cantidades de Equipos Evaluadas:

   - Las pruebas se llevaron a cabo para diferentes cantidades de equipos, desde 4 hasta 36.

4. Resultados:

   - Se presenta el tiempo promedio de ejecución para cada cantidad de equipos y cada algoritmo.

   - Los tiempos están expresados en segundos y se indican con un formato de ocho decimales.

5. Tendencias Observadas:

   - En general, se observa que el tiempo de ejecución aumenta a medida que la cantidad de equipos aumenta para ambos algoritmos.

   - La "solución ingenua" parece tener tiempos de ejecución más altos en comparación con el algoritmo "calendario" para todas las cantidades de equipos evaluadas.

   - Para ambas soluciones, se espera que el tiempo de ejecución aumente a medida que la cantidad de equipos crece, ya que la complejidad del problema probablemente aumenta con el tamaño del conjunto de equipos.

6. Análisis de Eficiencia:

   - Estos resultados sugieren que el algoritmo "calendario" puede ser más eficiente en términos de tiempo de ejecución en comparación con la "solución ingenua", especialmente a medida que la complejidad del problema aumenta con un mayor número de equipos.

7. Consideraciones Adicionales:

   - Es importante tener en cuenta que el rendimiento de los algoritmos puede depender de varios factores, como la implementación específica, las características del hardware y la eficiencia del código.

8. Conclusiones:

- Con base en estos resultados, se podría concluir que el algoritmo "calendario" es más eficiente en términos de tiempo de ejecución para generar calendarios en comparación con la "solución ingenua", al menos en el rango de cantidades de equipos evaluadas.

En resumen, esta prueba proporciona información valiosa sobre el rendimiento relativo de dos algoritmos de generación de calendarios en función del número de equipos involucrados.

Espacio:

Espacio por Algoritmo calendario

Número de equipos: 4, Uso de memoria promedio: 8.00000000 kilobytes

Número de equipos: 4, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 4, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 4, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 4, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 4, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 4, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 4, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 4, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 4, Uso de memoria promedio: 0.00000000 kilobytes

----

Número de equipos: 6, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 6, Uso de memoria promedio: 4.00000000 kilobytes

Número de equipos: 6, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 6, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 6, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 6, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 6, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 6, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 6, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 6, Uso de memoria promedio: 0.00000000 kilobytes

\--

\--

Número de equipos: 8, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 8, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 8, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 8, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 8, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 8, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 8, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 8, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 8, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 8, Uso de memoria promedio: 0.00000000 kilobytes

\--

\--

Número de equipos: 10, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 10, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 10, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 10, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 10, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 10, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 10, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 10, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 10, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 10, Uso de memoria promedio: 0.00000000 kilobytes

----

Número de equipos: 12, Uso de memoria promedio: 8.00000000 kilobytes

Número de equipos: 12, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 12, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 12, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 12, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 12, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 12, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 12, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 12, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 12, Uso de memoria promedio: 0.00000000 kilobytes

----

Número de equipos: 14, Uso de memoria promedio: 8.00000000 kilobytes

Número de equipos: 14, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 14, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 14, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 14, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 14, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 14, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 14, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 14, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 14, Uso de memoria promedio: 0.00000000 kilobytes

----

Número de equipos: 18, Uso de memoria promedio: 8.00000000 kilobytes

Número de equipos: 18, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 18, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 18, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 18, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 18, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 18, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 18, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 18, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 18, Uso de memoria promedio: 0.00000000 kilobytes

----

Número de equipos: 20, Uso de memoria promedio: 12.00000000 kilobytes

Número de equipos: 20, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 20, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 20, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 20, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 20, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 20, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 20, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 20, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 20, Uso de memoria promedio: 0.00000000 kilobytes

----

Número de equipos: 30, Uso de memoria promedio: 44.00000000 kilobytes

Número de equipos: 30, Uso de memoria promedio: 8.00000000 kilobytes

Número de equipos: 30, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 30, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 30, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 30, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 30, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 30, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 30, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 30, Uso de memoria promedio: 0.00000000 kilobytes

---

Número de equipos: 36, Uso de memoria promedio: 28.00000000 kilobytes

Número de equipos: 36, Uso de memoria promedio: 12.00000000 kilobytes

Número de equipos: 36, Uso de memoria promedio: 12.00000000 kilobytes

Número de equipos: 36, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 36, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 36, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 36, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 36, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 36, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 36, Uso de memoria promedio: 0.00000000 kilobytes

---

Espacio por Algoritmo solución ingenua

Número de equipos: 4, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 4, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 4, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 4, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 4, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 4, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 4, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 4, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 4, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 4, Uso de memoria promedio: 0.00000000 kilobytes

----

Número de equipos: 6, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 6, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 6, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 6, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 6, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 6, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 6, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 6, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 6, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 6, Uso de memoria promedio: 0.00000000 kilobytes

----

Número de equipos: 8, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 8, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 8, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 8, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 8, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 8, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 8, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 8, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 8, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 8, Uso de memoria promedio: 0.00000000 kilobytes

----

Número de equipos: 10, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 10, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 10, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 10, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 10, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 10, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 10, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 10, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 10, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 10, Uso de memoria promedio: 0.00000000 kilobytes

----

Número de equipos: 12, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 12, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 12, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 12, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 12, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 12, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 12, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 12, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 12, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 12, Uso de memoria promedio: 0.00000000 kilobytes

----

Número de equipos: 14, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 14, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 14, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 14, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 14, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 14, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 14, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 14, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 14, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 14, Uso de memoria promedio: 0.00000000 kilobytes

----

Número de equipos: 18, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 18, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 18, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 18, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 18, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 18, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 18, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 18, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 18, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 18, Uso de memoria promedio: 0.00000000 kilobytes

----

Número de equipos: 20, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 20, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 20, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 20, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 20, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 20, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 20, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 20, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 20, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 20, Uso de memoria promedio: 0.00000000 kilobytes

----

Número de equipos: 30, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 30, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 30, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 30, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 30, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 30, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 30, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 30, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 30, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 30, Uso de memoria promedio: 0.00000000 kilobytes

----

Número de equipos: 36, Uso de memoria promedio: 4.00000000 kilobytes

Número de equipos: 36, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 36, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 36, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 36, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 36, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 36, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 36, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 36, Uso de memoria promedio: 0.00000000 kilobytes

Número de equipos: 36, Uso de memoria promedio: 0.00000000 kilobytes

----

Cada sección corresponde a un conjunto de resultados para un algoritmo específico, y se presentan en un formato tabular con dos columnas principales: "Número de equipos" y "Uso de memoria promedio" en kilobytes.

Aquí hay algunas observaciones generales sobre los datos proporcionados:

Algoritmo "calendario":

Se presentan resultados para diferentes cantidades de equipos (4, 6, 8, 10, 12, 14, 18, 20, 30, 36).

Para la mayoría de las configuraciones, el "Uso de memoria promedio" es 0.00000000 kilobytes, indicando que el algoritmo no utiliza memoria significativa en esas pruebas.

Sin embargo, en algunos casos específicos (por ejemplo, equipos 12, 14, 18, 20, 30), se observa un "Uso de memoria promedio" de 8.00000000 o 12.00000000 kilobytes.

Algoritmo "sol_ingenua":

Similar al algoritmo anterior, se presentan resultados para diversas cantidades de equipos.

La característica común es que el "Uso de memoria promedio" es predominantemente 0.00000000 kilobytes, indicando un bajo uso de memoria en general.

En algunos casos (por ejemplo, equipos 20, 30, 36), se observa un uso de memoria promedio mayor, alcanzando hasta 44.00000000 kilobytes.

e) Análisis de los resultados del punto anterior. ¿Coincide el comportamiento de la implementación con la complejidad esperada de cada una de las soluciones?

A pesar de la elevada complejidad del algoritmo calendario, se ha observado que su rendimiento es más eficiente de lo inicialmente esperado.

f) Conclusiones y aspectos a mejorar.

Calendario

Desglose de Complejidad Temporal y Espacial:

Complejidad Temporal:

- Creación de las matrices calendario y partidos_jugados: O(n^2).
- Primer bucle for anidado: O(n^2).
- Creación de la lista equipos_contrincantes: O(n).
- Operación min: O(n).
- Operaciones de eliminación en equipos_restantes: O(n).
- Segundo bucle for anidado: O(n^2).
- Complejidad Total Temporal: O(n^3).

Complejidad Espacial:

- Matrices calendario y partidos_jugados: O(n^2).

Solución Ingenua:

Complejidad Temporal:

- Bucle principal: O(n).
- Bucle del equipo local: O(n).
- Operación de filtrado y eliminación: O(n).
- Bucle adicional al final: O(n^2).
- Complejidad Total Temporal: O(n^2).

Complejidad Espacial:

- Matrices calendario y partidos_jugados: O(n^2).

Aspectos para mejorar

- Estabilidad
- Manejo de Excepciones
- Pruebas Unitarias
- Eficiencia del Algoritmo
- Modularización