

Universidad del Valle
Fundamentos de Análisis y Diseño de Algoritmos
Mini-Proyecto
Febrero 23 de 2022

Profesor: Jesús Alexander Aranda

Monitor: Reynell Arkad Devji Quevedo

1. Introducción

El siguiente proyecto tiene como objetivo que los estudiantes del curso puedan:

- Analizar un problema relacionado con la realidad actual y su relación con los problemas vistos en clase.
- Emplear diferentes técnicas de programación para dar diferentes alternativas de solución.
- Analizar la eficiencia de las soluciones planteadas al problema.

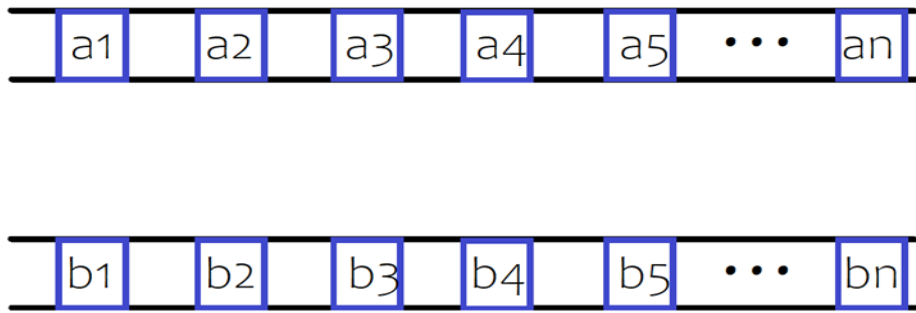
Optimizando los tiempos de producción en líneas de ensamblaje

Supongamos que existen una empresa de ensamblaje, la cual cuenta con un determinado número de líneas para a cabo el desarrollo de cierto producto, el cual pasa por un proceso que se puede descomponer en cierto número de trabajos a desarrollar de forma secuencial. Uno de los problemas que se podría considerar es cómo utilizar más de una línea de ensamblaje para minimizar los tiempos totales de ensamblaje del producto. Ese es el tema de este trabajo.

Con base en el anterior contexto se plantea a continuación un problema relacionado en el cual ustedes podrán plantear alternativas de solución según la eficiencia.

Caso concreto: Optimizar el número de total de horas para ensamblar un producto usando dos líneas de ensamblaje

Usted trabaja en una empresa de ensamblaje que tiene 2 líneas de ensamblado de la siguiente forma.

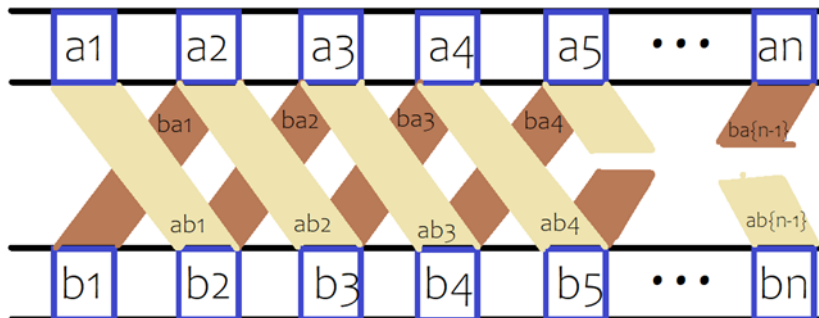


Cada línea tiene n puntos de ensamblado, el valor en cada punto representa el tiempo en horas que requiere la respectiva actividad de ensamblaje. Por ejemplo: a_1 representa el número de horas que necesita la línea de ensamblaje a para poder realizar la actividad 1, mientras b_1 representaría el número de horas para la misma actividad, pero en la línea b .

a_i : número de horas de la i -ésima actividad en la línea a .

b_i : número de horas de la i -ésima actividad en la línea b .

A pesar de ser 2 líneas separadas, estas pueden conectarse entre sí para realizar un trabajo de la mejor forma posible (en este caso de la forma más rápida) y no realizarlo necesariamente en una sola, representado de la siguiente forma.



Donde ba es un arreglo de tamaño $n-1$ que identifica el tiempo en horas de pasar de la línea (carril) b a la línea (carril) a en una posición posterior, y ab es un arreglo de tamaño $n-1$ que identifica el tiempo en horas de pasar de la línea (carril) a a la línea (carril) b en una posición posterior.

Por ejemplo:

ab_1 es el tiempo en horas para pasar del punto de ensamblaje 1 en la línea a al punto de ensamblaje 2 en la línea b (se asume que entonces ya se hizo la actividad 1).

ab2 es el tiempo en horas para pasar del punto de ensamblaje 2 en la línea a al punto de ensamblaje 3 en la línea b (se asume que entonces ya se hizo hasta la actividad 2).

ba1 es el tiempo en horas para pasar del punto de ensamblaje 1 en la línea b al punto de ensamblaje 2 en la línea a (se asume que entonces ya se hizo hasta la actividad 1)

ba2 es el tiempo en horas para pasar del punto de ensamblaje 2 en la línea b al punto de ensamblaje 3 en la línea a (se asume que entonces ya se hizo hasta la actividad 2)

En general, podemos decir que:

abi es el tiempo en horas para pasar del punto de ensamblaje i en la línea a al punto de ensamblaje i+1 en la línea b (se asume que entonces ya se hizo hasta la actividad i).

bai es el tiempo en horas para pasar del punto de ensamblaje i en la línea b al punto de ensamblaje i+1 en la línea a (se asume que entonces ya se hizo hasta la actividad i)

Dados los arreglos a, b, ab y ba; determine el mínimo tiempo en horas para realizar el ensamblaje completo pasando por los n puntos (ya sea en la línea a, la línea b o moviéndose entre ellas).

En concreto, se desea seleccionar en qué línea se va a realizar cada una de las n actividades (línea a o línea b) de tal forma que se minimice el tiempo total del ensamblaje, el cual se calcula sumando lo que costó en tiempo realizar cada actividad en la línea de ensamblaje seleccionada más los costos en tiempo de pasar de una línea a otra cada vez que haya ocurrido.

Ejemplo: Suponga que tenemos un producto que requiere de tres actividades de ensamblaje, para lo cual cuenta con la siguiente información sobre los tiempos en cada una de las dos líneas y para pasar de una línea a la otra:

Línea a:

2	2	6
---	---	---

Línea b:

3	5	3
---	---	---

Cambios ab:

1	1
---	---

Cambios ba:

4	2
---	---

En este caso la solución del problema sería

a	a	b
---	---	---

Lo que quiere decir que para ensamblar el producto se hará de la siguiente manera:

- Iniciar en la línea a desarrollando la actividad 1
- Continuar en la línea a para desarrollar la actividad 2
- Pasar de la línea a a la línea b para llevar a cabo la actividad 3
- Desarrollar en la línea b la actividad 3

Por lo anterior el tiempo de ensamblaje total en este caso sería

$$a_1 + a_2 + a_3 + b_3 = 2 + 2 + 1 + 3 = 8$$

Algunas aclaraciones:

Se puede iniciar en cualquier línea.

El número de horas de cada actividad en cada línea, así como los tiempos de desplazamientos entre líneas son enteros positivos. Suponga que estos valores siempre son menores o iguales a $10 \times \text{número de actividades para ensamblar un producto}$ (es decir, menor o igual a $10 \times n$)

Se puede terminar en cualquier línea.

Con base en lo anterior, se les pide lo siguiente:

1. **Resuelvan el problema del caso concreto por medio de dos algoritmos basados, parcialmente, en algoritmos o técnicas vistas en clase. (uno de ellos puede ser, aunque no obligatorio, usando programación dinámica)**

2.1. Entrada

Sus algoritmos van a recibir como entrada un archivo de texto que está organizado de la siguiente forma

```
n
a1 a2 ... an
b1 b2 ... bn
ab1 ab2 ... ab(n-1)
ba1 ba2 ... ba(n-1)
```

Donde la primera línea contiene un el número de actividades de ensamblaje (n), la siguiente línea representa los tiempos del ensamblaje en la línea a para cada actividad, la siguiente línea los tiempos del ensamblaje en la línea b para cada actividad, la siguiente línea los tiempos de desplazamiento de la línea a a la línea b y finalmente la línea con los tiempos de desplazamiento de la línea b a la línea a.

2.2. Salida

Se espera que la salida sea un archivo de texto como el siguiente:

```
n
tiempo_total
línea_punto1
línea_punto2
línea_punto3
..
línea:punton
```

Done la primera línea indica el número de actividades para ensamblar el producto, la siguiente línea el tiempo total de la solución y las siguientes n líneas indican en que línea (a o b) se llevo a cabo cada una de las actividades.

3. Análisis

- **Eficiencia en tiempo:** Para cada una delas alternativas se deberá calcular su complejidad. Compruebe su análisis con las medidas de tiempo de ejecución tomadas para diferentes entradas y diferentes tamaños. Esto implica que debe generar un número importante de ejemplos de pruebas y guardarlas en un archivo, para poder probar con las diferentes alternativas usando los mismos datos. También se deben comparar los tiempos tomados por las diferentes alternativas y sacar conclusiones.

Se espera que se envíen las instancias de prueba (archivos de entrada y salida); mayor número de pruebas es mejor. Si hay dificultades con generarlas, enviar al menos 5 instancias de prueba. Si no es posible, explicar en el documento las dificultades que se tuvieron.

¿Qué se espera en los resultados y pruebas?

Se espera generar pruebas con diferentes tamaños y contrastar con los tiempos de respuesta, esto se pueden representar por medio de una tabla y una función en un plano cartesiano, por ejemplo. Finalmente se puede contrastar la complejidad teórica y la experimental obtenida a partir de las pruebas.

4. Grupos de trabajo

El proyecto puede ser desarrollado por grupos de máximo 4 personas.

5. Entrega

Las entregas consisten en un archivo empaquetado y comprimido. El nombre del archivo debe ser escrito de la siguiente forma: **ProyectoApellido1Apellido2Apellido3Apellido4.zip**. Debe contener los siguientes elementos:

- Código fuente con la aplicación.
- Archivos de entrada y salida utilizados
- Archivo proyecto.pdf con:
 1. Descripción de la idea, pseudocódigo y análisis de complejidad de las soluciones.
 2. Análisis de la complejidad temporal de las soluciones implementadas.
 3. Detalles principales de la implementación.
 4. Descripción y análisis de las pruebas realizadas.
 5. Conclusiones y aspectos para mejorar.

La fecha máxima de entrega es el 22 de marzo a las 23:55 p.m por medio del campus virtual.