

Entregable 2

La imprenta

EI1022/MT1022 - Algoritmia 2022/2023

Fecha de entrega: miércoles 2 de noviembre de 2022

Contenido

1. El problema.....	1
2. Ficheros de partida.....	2
3. Entrega en el aula virtual	5
4. Calificación del entregable.....	6

1. El problema

En una imprenta necesitan imprimir n folletos rectangulares de diferentes tamaños. Disponen de una impresora que trabaja con hojas de papel cuadradas de $M \times M$ unidades. Desean imprimir los folletos utilizando el menor número posible de hojas. El alto y ancho de cada folleto se representan mediante enteros entre 1 y M .

Así pues, el problema es distribuir los n folletos en k hojas de papel, con el objetivo de minimizar k . Para simplificar el problema consideraremos que los folletos deben mantener su orientación (no pueden girarse).

Implementa un programa de línea de órdenes, `entregable2.py`, que reciba, por la entrada estándar, el entero M y los n folletos en el formato que se especifica en el apartado 1.1. Como resultado de su ejecución, el programa mostrará sus resultados por la salida estándar en el formato que se detalla en el apartado 1.2.

Recuerda, al tratarse de un algoritmo voraz la decisión para cada folleto sólo debe tomarse una vez. Además, para tener un coste temporal razonable, tu algoritmo no debería depender de la superficie total de los folletos ni de la superficie de las hojas cuadradas de la imprenta.

1.1. Formato de la entrada

Una instancia del problema consistirá en un fichero de líneas de texto. La primera línea contendrá el entero M y a continuación habrá una línea por folleto, cada una de las cuales contendrá tres enteros, el número del folleto, su anchura y su altura, respectivamente. Los números de folleto empiezan en uno y están consecutivos en el fichero.

Se garantiza que la anchura y altura de cada folleto será menor que M .

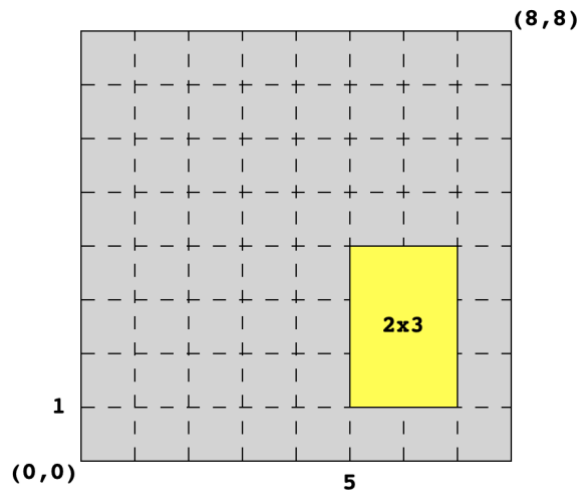
Por ejemplo, veamos el contenido del fichero `'small_1000_03_02.i'`:

```
1000
1 900 333
2 333 800
3 250 200
```

Para esta instancia, tenemos que $M = 1000$, es decir, las hojas que utiliza la imprenta serán de 1000x1000. También vemos que hay que imprimir tres folletos. El primero de los folletos tiene anchura 900 y altura 333, el segundo 333 y 800, y el tercero 250 y 200.

1.2. Formato de la salida

La siguiente figura representa un folleto de anchura 2 y altura 3 colocado en la posición (5, 1) en una hoja de la imprenta de tamaño 8x8:



El programa deberá mostrar, por la salida estándar, una línea de texto por folleto, cada una con cuatro enteros separados por un blanco. Estos cuatro enteros son, por orden:

- Número del folleto: Leído del fichero de instancia. Cuentan desde 1.
- Número de la hoja en la que va colocado dicho folleto: Cuentan desde 1 y no puedes dejar ninguna hoja vacía.
- Posición horizontal en la hoja donde empieza el folleto (considerando la posición 0 del eje horizontal como el lado izquierdo del papel). Para el ejemplo del dibujo sería un 5.
- Posición vertical en la hoja donde empieza el folleto (considerando la posición 0 del eje vertical como el lado inferior del papel). Para el ejemplo del dibujo sería un 1.

Como ejemplo, una posible salida para la instancia 'small_1000_03_02.i' sería:

```
1 1 0 0
3 1 0 333
2 2 0 0
```

2. Ficheros de partida

En el aula virtual disponéis de una carpeta con el siguiente contenido:

- `public_test.zip`: un archivo comprimido que contiene la carpeta `public_test` con un conjunto de instancias de prueba.
- `entregable2.py`: el programa que debéis de utilizar como punto de partida.
- `entregable2_viewer.py`: un programa que importa `entregable2.py` y lo utiliza para resolver una instancia y mostrar la solución de forma gráfica por pantalla.
- `entregable2_test.py`: programa que importa `entregable2.py` y lo utiliza para validar tanto la solución obtenida como el tiempo que la función `process()` ha necesitado para obtenerla.

Los siguientes apartados explican con detalle estos contenidos.

2.1. Ficheros de prueba (public_test.zip)

Al descomprimir el fichero public_test.zip obtendremos la carpeta 'public_test', que contiene:

- Ficheros 'large_<s>_<n>_<r>.i': Diez instancias de prueba, del mismo tamaño que las instancias privadas con las que se probará el programa.
- Ficheros 'small_<s>_<n>_<r>.i': Diez instancias más sencillas para ayudar en la depuración.

A partir del nombre de un fichero de instancia, puedes averiguar:

- <s>: El lado de las hojas cuadradas de la imprenta.
- <n>: El número de folletos que contiene la instancia.
- <r>: El número máximo de hojas que puedes utilizar en tu solución.

2.2. Fichero principal (entregable2.py)

Fichero con la estructura del programa. Debéis utilizarlo con las siguientes restricciones:

- Podéis añadir funciones o clases adicionales, pero **no debéis modificar nada del programa principal**. Modificar el programa principal supondrá un cero en la calificación del entregable. Tampoco podéis modificar la firma (el tipo) de las tres funciones que se utilizan en el programa principal.
- Podéis importar módulos estándar de Python.
- Podéis importar módulos de la biblioteca `algorithmia`, pero sin cambiar la ruta al importarlos.

Veamos la estructura de entregable2.py y las funciones que debéis implementar.

Para empezar, y por comodidad, el programa define los tipos `Leaflet` y `LeafletPos` (folleto en inglés es *leaflet*):

```
Leaflet = tuple[int, int, int]          # (num_folleto, anchura, altura)
LeafletPos = tuple[int, int, int, int]  # (num_hoja, num_folleto, posx, posy)
```

El programa principal utiliza la estructura de tres funciones vista en clase:

```
paper_size, leaflet_list = read_data(sys.stdin)
leafletpos_list = process(paper_size, leaflet_list)
show_results(leafletpos_list)
```

Este entregable consiste en implementar las tres funciones que aparecen en el programa principal:

- **read_data(f: TextIO) -> tuple[int, list[Leaflet]]**
Entrada: El fichero con la instancia (ojo, `f` no es un nombre de fichero, es un fichero).
Salida: Una tupla de dos elementos, el primero es el lado de las hojas cuadradas de la imprenta; el segundo elemento de la tupla es una lista con los folletos, donde cada folleto es una tupla `Leaflet`.
- **process(paper_size: int, leaflet_list: list[Leaflet]) -> list[LeafletPos]**
Entrada: Dos parámetros, los mismos que forman la tupla de salida de la función `read_data`.
Salida: Una lista de tuplas de tipo `LeafletPos`, donde cada tupla contiene la información con la colocación de un folleto.
- **show_results(leafletpos_list: list[LeafletPos])**
Entrada: Un parámetro, el mismo que devuelve la función `process`.
Salida: No devuelve nada. Solo muestra texto por la salida estándar siguiendo el formato que se detalla en el apartado 1.2.

Sigue con detalle estos pasos para ejecutar entregable2.py desde la línea de órdenes:

1. Si no está instalada ya, instala la biblioteca `algoritmia`. Se ha explicado cómo hacerlo en la segunda sesión de prácticas (la información está disponibles en el aula virtual).
2. Abre un terminal y ve al directorio donde está el fichero `entregable2.py` y la carpeta `public_test` (usa `cd`).
3. Ya puedes lanzar tu programa:

```
python3.10 entregable2.py < public_test/small_1000_03_02.i
```

que deberá mostrar por pantalla una línea por folleto, según se detalla en el apartado 1.2.

2.3. El validador de soluciones (`entregable2_test.py`)

Una solución es válida si todos los folletos están colocados en alguna hoja de la imprenta, sin que sobresalgan de la hoja y sin que se solapen entre sí. Además, las hojas deben estar numeradas consecutivamente y no debe haber hojas vacías.

El validador de soluciones (`entregable2_test.py`) es un programa que importa tu fichero `entregable2.py` y lo utiliza para resolver una instancia del problema. Para utilizarlo, abre un terminal, ve al directorio donde están `entregable2_test.py`, la carpeta `public_test` y tu `entregable2.py`, y ejecuta la orden:

```
python3.10 entregable2_test.py public_test/small_1000_03_02.i
```

El resultado de la ejecución para una instancia puede ser:

- **OK:** La solución es válida y se ha obtenido dentro del límite de tiempo. Se muestra una salida como esta:

```
INSTANCE: public_test/small_1000_03_02.i
RESULT: OK
NUMPAGES: OK - 2 pages (<= 2 pages)
USEDTIME: OK - 0.00 sec (<= 3 sec)
```

- **ERROR_NUMPAGES:** La solución es válida, pero se ha superado el número de hojas máximo. Se muestra una salida como esta:

```
INSTANCE: public_test/small_1000_03_02.i
RESULT: ERROR_USED_PAGES
NUMPAGES: ERROR - 3 pages (> 2 pages)
USEDTIME: OK - 0.00 sec (<= 3 sec)
```

- **ERROR_TIMEOUT:** La solución es válida, pero se ha superado el límite de tiempo de tres segundos. Se muestra una salida como esta:

```
INSTANCE: public_test/small_1000_03_02.i
RESULT: ERROR_TIMEOUT
NUMPAGES: OK - 2 pages (<= 2 pages)
USEDTIME: ERROR - 3.42 sec (> 3 sec)
```

- **ERROR_INVALID_SOLUTION.** Tu programa termina correctamente, pero la solución obtenida no es válida. Se muestra el motivo por el que no es válida.
- **ERROR_CHECK_FAILED.** No se ha producido ningún error de ejecución, pero tu programa no ha pasado alguna verificación intermedia. Se muestra el problema detectado.
- **ERROR_EXCEPTION_LAUNCHED.** Se ha producido un error de ejecución que ha lanzado una excepción. Se muestra la excepción que se ha lanzado e información de traza para depuración.

Hay otro resultado posible: que alguna llamada a tu programa no termine (en un plazo razonable) y tengas que pulsar `Ctrl-C` para cancelar la ejecución del validador.

Opcionalmente, si lo prefieres, puedes probar tu programa con todas las instancias con una única orden:

```
python3.10 entregable2_test.py public_test/*.i
```

2.4. El visor de soluciones (entregable2_viewer.py)

Este programa importa tu fichero entregable2.py por lo que ambos deben estar en el mismo directorio.

Abre un terminal y ve al directorio donde están entregable2_viewer.py, entregable2.py y la carpeta public_test. Ejecuta la siguiente orden (en Windows utiliza la barra invertida, '\'):

```
python3.10 entregable2_viewer.py public_test/small_1000_03_02.i
```

Se abrirá una ventana con tu distribución de folletos en la primera hoja. Puedes moverte entre hojas con las fechas del teclado:

- →: pasa a la hoja siguiente. Vuelve a la primera si estás en la última.
- ←: pasa a la hoja anterior. Va a la última si estás en la primera.
- ↑: avanza 100 hojas, sin pasar de la última.
- ↓: retrocede 100 hojas, sin pasar de la primera.

3. Entrega en el aula virtual

La entrega consistirá en un subir **dos** ficheros concretos a la tarea correspondiente del aula virtual, **solo debe subirlos uno de los miembros del grupo**. Estos son los dos ficheros:

- **entregable2.py**: El fichero con el código del entregable.
- **alxxxxxx_alyyyyyy.txt**: Un fichero de texto que deberá contener el nombre, el número de DNI y el al##### de cada miembro del grupo (el formato del contenido es libre). Evidentemente, en el nombre del fichero tenéis que reemplazar alxxxxxx y alyyyyyy por los correspondientes a los dos miembros del grupo.

Si el grupo es unipersonal, el fichero de texto deberá llamarse **alxxxxxx.txt**, reemplazando alxxxxxx por el que corresponda.

Tu entrega debe cumplir estas restricciones (cada restricción no cumplida quita un punto del entregable):

- Los nombres de los dos ficheros deben utilizar únicamente minúsculas.
- El separador utilizado en el nombre de fichero 'alxxxxxx_alyyyyyy.txt' es el guion bajo ('_'), no utilices un menos ('-') ni ningún otro carácter similar.
- Debes poner correctamente las extensiones de los dos archivos ('.py' y '.txt'). Si utilizas Windows y tienes las extensiones de archivo ocultas (que es la configuración por defecto de Windows) es posible que envíes ficheros con doble extensión, evítalo: configura tu Windows para que muestre las extensiones de los archivos conocidos.
- No subas ningún fichero ni directorio adicional.
- Nada de comprimir los archivos y subir un zip, rar o similar.

4. Calificación del entregable

El entregable se calificará utilizando unas pruebas privadas que se publicarán junto con las calificaciones.

Las pruebas privadas consistirán en diez instancias con los mismos tamaños que las instancias 'large' de las pruebas públicas: 'large_<s>_<n>_<r>.i'.

El ordenador con el que se medirán los tiempos de ejecución será `lynx.uji.es`. Un ordenador al que todos tenéis acceso y en el que podéis ejecutar el programa `entregable2_test.py` que os proporcionamos.

Para considerar superada una instancia 'large_<s>_<n>_<r>.i', vuestro método `process()` no deberá tardar más de tres segundos y la solución obtenida no podrá necesitar un número de hojas mayor que `<r>`.

Así pues, el programa puede estar mal de tres formas diferentes:

- Tener uno o más errores y funcionar mal con algunas instancias (o con todas).
- No tener errores, pero sobrepasar el tiempo máximo de tres segundos al resolver la instancia.
- No tener errores, pero devolver un número de hojas superior al máximo permitido por la instancia.

Estos problemas pueden detectarse utilizando las pruebas públicas, aunque superar las pruebas públicas no garantiza superar las privadas, sobre todo si la implementación se ha 'ajustado' específicamente para superar las públicas.

4.1. Errores graves en el entregable

Obtendréis directamente una puntuación de cero en el entregable si modificáis el programa principal de `entregable2.py` o los tipos de cualquiera de las funciones que utiliza.

Se penalizará también con un cero si vuestro programa no lee las instancias de la entrada estándar, tal y como se indica en el apartado 1.1.

También se penalizará que la salida no respete el formato que se especifica en el apartado 1.2. Una salida errónea puede tener dos causas:

- Implementación que no respeta el formato especificado. La penalización consistirá en quitar **dos puntos** a la nota del entregable.
- El programa tiene algún `print` olvidado en el código que se ejecuta durante las pruebas. La penalización consistirá en quitar **un punto** a la nota del entregable.

Por último, también se penaliza con un punto cada restricción incumplida en la entrega al aula virtual (ver el apartado 3).

Revisadlo con detenimiento antes de entregar (todos los miembros del grupo).

4.2. Penalización por copia

En caso de detectarse una copia entre grupos, se aplicará la normativa de la universidad: la calificación de la evaluación continua (60 % de la nota final) será de cero en la primera convocatoria para todos los miembros de los grupos involucrados.