Entregable 5
El puente

EI1022/MT1022 - Algoritmia 2022/2023

Fecha de entrega: viernes 23 de diciembre de 2022

Contenido

1. El problema

Para cruzar un río tenemos que utilizar un puente formado por N tablones. Cada tablón tiene una cantidad de puntos (positiva o negativa) que obtenemos al pisarlo. Inicialmente estamos encima del tablón 0, y queremos llegar al otro lado del puente, más allá del último tablón. Podemos efectuar un máximo de M movimientos (saltos) y en cada uno de ellos saltar hasta K tablones. Con esta idea, desde un tablón i podemos movernos a un tablón en el rango $[i+1\ ..\ i+K]$. Si el destino del salto es mayor o igual a N, salimos del puente y llegamos al otro lado. La puntuación del recorrido es la suma de los puntos de los tablones utilizados.

Se cumple que $M * K \geq N \geq 1$ por lo que siempre podremos cruzar el puente.

Queremos averiguar la puntuación máxima que podemos conseguir, así como los tablones que tenemos que pisar para obtenerla.

Se pide:

- Implementar un programa de línea de órdenes, `entregable5.py`, que reciba, por la entrada estándar, una instancia en el formato que se especifica en el apartado 1.1. El programa deberá mostrar por la salida estándar la solución que maximiza la puntuación obtenida, en el formato que se especifica en el apartado 1.2.
- El programa deberá resolver el problema mediante programación dinámica.

1.1. Formato de la entrada

Una instancia del problema consistirá en un fichero de líneas de texto.

La primera línea contiene dos enteros separados por un blanco, el primero es la distancia máxima de salto (K) y el segundo el número máximo de saltos (M).

Las líneas siguientes contienen cada una un entero con los puntos asignados a cada tablón, empezando por el 0 y en orden. Llamaremos T a este vector de puntos.

Como ejemplo de instancia, veamos el contenido del fichero 'instance_01_00645.i':

```
3 5
-536
-242
970
-232
-742
-605
443
```

Como vemos, contiene la instancia: $K = 3$, $M = 5$ y $T = [-536, -242, 970, -232, -742, -605, 443]$.

1.2. Formato de la salida

La salida consistirá en varias líneas de texto. La primera contendrá la puntuación obtenida. El resto de lineas contendrán los índices de los tablones que pisamos (en orden creciente).

Como ejemplo de solución veamos el contenido del fichero 'instance_01_00645.o':

```
645
0
2
3
6
```

El contenido indica que la puntuación máxima es de 645 puntos y que la podemos obtener pisando los tablones con los índices 0, 2, 3 y 6. Si sumamos los puntos en dichos tablones obtenemos los 645 puntos (536+970-232+443). Como podemos observar, la solución efectúa cuatro saltos: del 0 al 2, del 2 al 3, del 3 al 6 y del 6 salta fuera del puente.

2. Ficheros de partida

En el aula virtual disponéis del fichero `entregable5-puente.zip`, que al descomprimirlo crea una carpeta con el siguiente contenido:

- El fichero de partida: `entregable5.py`
- El validador de soluciones: `entregable5_test.py`
- La carpeta con las instancias de prueba: `public_test`

2.1. El fichero de partida: `entregable5.py`

El fichero sigue el esquema de la asignatura. Las funciones `read_data` y `show_results` ya están implementadas, únicamente tenéis que implementar la función `process`, cuya firma es:

```python
def process(K: int, M: int, T: list[Score]) -> tuple[Score, list[Decision]]:
```

donde:

```python
Score = int     # Puntos
Decision = int  # Índice en el vector T
```

2.2. El validador de soluciones: `entregable5_test.py`

El validador de soluciones (`entregable5_test.py`) es un programa que importa tu fichero `entregable5.py` y lo utiliza para resolver una instancia del problema. Para utilizarlo, abre un terminal, ve al directorio donde están `entregable5_test.py`, la carpeta `public_test` y tu `entregable5.py`, y ejecuta la orden:

```
python3.10 entregable5_test.py public_test/instance_01_00645.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/instance_01_00645.i
RESULT: OK
  SCORE:    OK – 645 points
  USEDTIME: OK - 0.00 sec (<= 1 sec)
```

- **ERROR_SCORE**: La puntuación no es la óptima:

```
INSTANCE: public_test/instance_01_00645.i
RESULT: ERROR_SCORE
  SCORE:    ERROR – 742 points
  USEDTIME:    OK - 0.00 sec (<= 1 sec)
```

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

```
INSTANCE: public_test/instance_01_00645.i
RESULT: ERROR_TIMEOUT
  SCORE:       OK – 645 points
  USEDTIME: ERROR - 1.45 sec (> 1 sec)
```

- **ERROR_INVALID_SOLUTION**. No se ha producido ningún error de ejecución, 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.

2.3. La carpeta con las instancias de prueba: `public_test`

La carpeta '`public_test`' contiene diez instancias de prueba y sus respectivas soluciones.

El nombre de cada instancia sigue el patrón '`instancia_<n>_<m>.i`', donde <n> es el número de instancia y <m> es un entero que indica la puntuación máxima que se puede conseguir.

Las soluciones tienen el mismo nombre que las instancias pero reemplazando la extensión '`.i`' por '`.o`'. Vuestras soluciones pueden diferir en los saltos, pero deben obtener la misma puntuación (la óptima).

3. Entrega en el aula virtual

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

- **`entregable5.py`**: El fichero con el código principal 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.

Vuestra 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.
- Debéis poner correctamente las extensiones de los dos archivos ('.py' y '.txt'). Si utilizáis Windows y tenéis las extensiones de archivo ocultas (que es la configuración por defecto de Windows) es posible que enviéis ficheros con doble extensión, evitadlo: configurad Windows para que muestre las extensiones de los archivos conocidos.
- No subáis ningún fichero ni directorio adicional.

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 similares a las de la carpeta `public_test`.

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 `entregable5_test.py` que os proporcionamos.

Para considerar superada una instancia '`instancia_<n>_<m>.i`', vuestro método `process` no deberá tardar más de un segundo y la solución obtenida deberá conseguir `<m>` puntos.

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

- Tener uno o más errores y funcionar mal con algunas instancias (o con todas).
- Funcionar correctamente, pero sobrepasar el tiempo máximo de un segundo al resolver 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 `entregable5.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:

- Una 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.