

Tarea 1

Profesor/es: Juan Pablo Castillo
Ayudantes: Nangel Coello, Carlos Lagos
`nangel.coello@usm.cl`
`carlos.lagosc@usm.cl`

Fecha de Inicio: 30 de abril, 2024
Fecha de Entrega: 18 de mayo, 2024

Reglas del Juego

La presente tarea debe hacerse en grupos de 3 personas. Toda excepción a esta regla debe ser conversada con el ayudante **ANTES** de comenzar la tarea. No se permiten de ninguna manera grupos de más de 3 personas. Pueden usarse los lenguajes de programación C, C++, Python, y Java.

Problema 1: Fuerza Bruta Recursiva

Bob un destacado constructor, estaba desempleado y en busca de empleo. Un día recibió una llamada de una empresa reconocida internacionalmente llamada “ArMAZON Construction” para agendar una entrevista. Acudió a la entrevista, durante la cual se le realizó una prueba técnica. En esta prueba, se le solicitó construir una pared de longitud L y sin restricción de altura, con una secuencia $S[1..n] = \{l_1, l_2, \dots, l_n\}$ de ladrillos de altura 1 y de longitudes l_i , donde $l_i \leq L$ para $i = [1..n]$. La prueba para Bob no parecía ser difícil, pero para construir la pared se deben seguir las siguientes reglas:

1. Se comienza construyendo la fila más cercana al piso y de izquierda a derecha.
2. Cada fila de la pared es de la misma altura (altura 1).
3. En cada fila se puede insertar cualquier cantidad de ladrillos siempre y cuando $\sum_{k=i}^j l_k \leq L$, para $i \leq j$.
4. Se pueden dejar espacios entre los ladrillos.
5. No es necesario que los ladrillos tengan soporte debajo de ellos.
6. Los ladrillos deben ser colocados en el mismo orden de la secuencia $S[1..n]$.
7. Los ladrillos deben estar lo más uniformemente posible distribuidos en la pared.

Para resolver el problema Bob debe encontrar la mínima suma de los gaps¹ al cuadrado posible que quedan en la pared de largo L , utilizando la secuencia $S[1..n]$ de ladrillos.

¹Un gap es el espacio que queda sin cubrir en cada fila de la pared.

Por ejemplo, para construir una pared de $L = 6$ con la secuencia de ladrillos $S[1..9] = \{2, 3, 1, 4, 6, 2, 5, 1, 5\}$, la mejor solución es construir la pared de la siguiente forma:

Fila 6	1	5
Fila 5	5	
Fila 4	2	
Fila 3	6	
Fila 2	1	4
Fila 1	2	3

Note que, la mínima suma de los gaps que quedan en la pared al cuadrado es:

$$1^2 + 1^2 + 0^2 + 4^2 + 1^2 + 0^2 = 19.$$

Formato de Entrada

La entrada es leída desde la entrada standard, y contiene varios casos de prueba. Por cada caso hay un entero L que indica la longitud de la pared a construir y n que indica la cantidad de ladrillos. En la siguiente línea n enteros que representan las longitudes de los ladrillos (separadas por un único espacio). En este caso $1 \leq n \leq 10000$ y $0 < L \leq 10^6$. La entrada es terminada por EOF.

Un ejemplo de entrada es el siguiente:

```
6 9
2 3 1 4 6 2 5 1 5
10 12
2 3 1 4 6 2 5 1 5 9 1 10
```

Hint: para probar su programa de una mejor manera, ingrese los datos de entrada con el formato indicado en un archivo de texto (por ejemplo, el archivo `input-1.dat`). Luego, ejecute su programa desde la terminal, redirigiendo la entrada standard como a continuación, evitando tener que entrar los datos manualmente cada vez que prueba su programa:

```
./problema1 < input-1.dat
```

Formato de Salida

La salida debe mostrarse a través de la salida standard. Para cada caso de prueba, se debe mostrar una línea que contiene la mínima suma de los gaps al cuadrado posible que quedan en la pared de largo L , utilizando la secuencia de ladrillos.

La salida correspondiente a la entrada mostrada anteriormente es:

```
19
41
```

Problema 2: Programación Dinámica

Resuelva el Problema 1 usando la técnica de programación dinámica. Los formatos de entrada y salida son los mismos que en el Problema 1.

Entrega de la Tarea

La entrega de la tarea debe realizarse enviando un archivo comprimido llamado

`tarea1-apellido1-apellido2-apellido3.tar.gz`

(reemplazando sus apellidos según corresponda), o alternatively usando formato zip, en el sitio Aula USM del curso, a más tardar el día 18 de mayo, 2024, a las 23:59:00 hrs (Chile Continental), el cual contenga:

- El plazo máximo de entrega es de a lo más 5 días desde la fecha original de entrega (18 de mayo, 2024). Por cada día (o **fracción**) de atraso se descontarán 20 puntos de la nota de la tarea.
- Los archivos con el código fuente necesarios para el funcionamiento de la tarea.
- `NOMBRES.txt`, Nombre y ROL de cada integrante del grupo. También se debe indicar qué hizo cada integrante del grupo.
- `README.txt`, Instrucciones de compilación en caso de ser necesarias.
- `Makefile`, Instrucciones para compilación automática, en caso de ser necesarias.