

El problema de la muralla

NOTA: Si usted está leyendo este documento sin haber extraído el compactado que se le entregó, ciérrelo ahora, extraiga todos los archivos en el escritorio, y siga trabajando desde ahí. Es un error común trabajar en la solución dentro del compactado, lo cual provoca que los cambios no se guarden. Si usted comete este error y entrega una solución vacía, no tendrá oportunidad de reclamar.

Corre el siglo *XIII* y China enfrenta un desafío: resistir ante las hordas mongolas comandas por Temuyín, más conocido como Genghis Khan. Para ello deben reforzar sus líneas de defensa comenzando por la Gran Muralla China que se encuentra debilitada tras un asedio. Para esto, China le ha pedido ayuda dada la excelencia de la brigada de constructores que usted dirige.

El pedazo de muralla más sensible se encuentra dividido en n secciones de longitudes conocidas $\{p_1, p_2, \dots, p_n\}$. Usted dispone de k constructores que deben repararlas todas. En vistas de evitar el desplazamiento de los constructores por la vasta muralla, es necesario que estos se encarguen **solamente de reparar secciones contiguas**. A cada constructor le toma una unidad de tiempo reparar una unidad de muralla.

Por ejemplo, un constructor puede reparar en total las secciones $\{2, 3, 4\}$ que le tomará $p_2 + p_3 + p_4$ unidades de tiempo o solo la sección $\{1\}$ que le tomará p_1 unidades de tiempo, pero no las secciones $\{2, 4, 5\}$.

Usted debe garantizar la reparación de **todas las secciones en el menor tiempo posible**.



Genghis Khan (1162-1227)



Gran Muralla China

Ejemplo

Supongamos que tenemos 3 constructores y las longitudes de las secciones son $\{8, 1, 4, 9, 3\}$. Si al primer constructor le es asignado las secciones $\{1, 2\}$, al segundo las secciones $\{3, 4\}$ y al tercero la sección $\{5\}$, estos tardarán en reparar la muralla un total de 13 unidades de tiempo, dado que todos los constructores pueden comenzar a reparar sus secciones simultáneamente. En este caso el primero constructor demora

9 = (8 + 1) unidades de tiempo en reparar sus secciones, el segundo demora 13 = (4 + 9) y el tercero 3.

Note que esta distribución no es óptima, pues con la distribución { 1 } { 2, 3 } { 4, 5 } demorarían 12 unidades de tiempo.

Para ello, debe haber recibido junto a este documento una solución de Visual Studio con dos proyectos: una biblioteca de clases (*Class Library*) y una aplicación de consola (*Console Application*). Usted debe completar la implementación del método *Repara* en el *namespace* *Weboo*. Examen que está dentro de la clase *Muralla*.

NOTA: Todo el código de la solución debe estar en este proyecto (biblioteca de clases), pues es el único código que será evaluado. Usted puede adicionar todo el código que considere necesario, pero no puede cambiar los nombres del namespace, clase o método mostrados. De lo contrario, el probador automático fallará.

Ejemplos

```
int[] secciones1 = { 10, 10, 10, 10 };
int constructores1 = 2;
/* Acá se puede dividir las secciones en 2 grupos { 1, 2 } y { 3, 4 } y a cada constructor
le tomará 20 unidades de tiempo reparar la muralla, => el óptimo es 20 */

int[] secciones2 = { 10, 20, 30, 40 };
int constructores2 = 2;
/* Acá se puede dividir las secciones en 2 grupos { 1, 4 } y { 2, 3 } y a cada constructor
le tomará 50 unidades de tiempo reparar la muralla, => el óptimo es 50.
¡¡¡FALSO!!! Recuerde la restricción del problema: "(...) es necesario que estos se
encarguen solamente de reparar secciones contiguas".
Por lo que se debe dividir en { 1, 2, 3 } y { 4 } => el óptimo es 60 */
```

NOTA: Los casos de prueba que aparecen en este proyecto son solamente de ejemplo. Que usted obtenga resultados correctos con estos casos no es garantía de que su solución sea correcta y de buenos resultados con otros ejemplos. De modo que usted debe probar con todos los casos que considere convenientes para comprobar la validez de su implementación.