

URL VIDEOS

Explicación enunciado del problema: <https://youtu.be/AdlkczpxEFw>

Explicación código: <https://youtu.be/LOhvoWDj278>

Análisis de complejidad: <https://www.youtube.com/watch?v=l7X4zluGxkw>

ENUNCIADO DEL PROBLEMA

Vladislav escribe en la pizarra los números enteros del 1 al n, ambos inclusive. Después sustituye cada número entero por la suma de sus cifras.

¿Cuál es ahora la suma de los números del tablero?

Por ejemplo, si n=12 entonces inicialmente los números del tablero son:

1,2,3,4,5,6,7,8,9,10,11,12.

Después de la sustitución, los números pasan a ser:1,2,3,4,5,6,7,8,9,1,2,3.

La suma de estos números es $1+2+3+4+5+6+7+8+9+1+2+3=51$. Por lo tanto, para n=12 la respuesta es 51.

Entrada

La primera línea contiene un número entero t ($1 \leq t \leq 10^4$) - el número de casos de prueba.

La única línea de cada caso de prueba contiene un único entero n ($1 \leq n \leq 2 \cdot 10^5$) - el mayor número que escribe Vladislav.

Salida

Para cada caso de prueba, salida de un solo número entero - la suma de los números al final del proceso.

**URL DE GITLAB O GITHUB PROYECTO EN FORMA PRIVADA, USUARIO madarme@ufps.edu.co, con
rol mantener**

<https://gitlab.com/ejercicios-proyecto-ada/ejercicio-6-vlad-and-a-sum-of-digits>

MÉTODO 1

```
public int[] metodoCandido(int[] casos) {
    int[] result = new int[casos.length];
    for (int i = 0; i < casos.length; i++) {
        Secuencia<Integer> vec2 = new Secuencia<>(casos[i] + 1);

        for (int j = 0; j <= casos[i]; j++) {
            vec2.insertar(0);
        }

        for (int k = 1; k <= casos[i]; k++) {
            int num = k;
            int suma = 0;
            while (num > 0) {
                suma += num % 10;
                num /= 10;
            }

            suma += vec2.get(k - 1);
            vec2.set(k, suma);
        }
        result[i] = vec2.get(casos[i]);
    }

    return result;
}
```

Eficacia	<p>El método es eficaz, puesto a que tiene una solución para todas las entradas válidas ingresadas.</p>
	<p>Instancia 1:</p> <p>TAMAÑO SECUENCIA: 5 SECUENCIA: {86963, 148283, 43246, 51296, 173338}</p> <p>Resultado de la invocación:</p> <p>1890222 3252438 839512 1023225 3864574</p>
	<p>Instancia 2:</p> <p>TAMAÑO SECUENCIA: 50 SECUENCIA: {90360, 190591, 150541, 4200, 71195, 17089, 144093, 158220, 124051, 121644, 38068, 83347, 184841, 49410, 86743, 119813, 192090, 11069, 21753, 57198,</p>

87343,	191400,	133473,	100410,	143995,	83850,	188665,
103866,	35793,	190908,	52469,	167299,	158816,	190748,
23005,	139509,	23948,	148186,	131577,	162649,	41705,
13424,	109657,	105572,	53462,	26158,	183166,	167833,
83316,	147827}					

Resultado de la invocación:

1986858
4332664
3309211
62706
1493898
303985
3151653
3489138
2712700
2666520
731269
1794772
4167531
983133
1883386
2634621

4369866
195165
396891
1154769
1900054
4352115
2913873
2254701
3149866
1807599
4277089
2310987
678717
4340907
1046295
3711700
3505647
4336545
419545
3055725
436698
3250135
2876889

3595375
809487
232620
2430721
2342352
1067097
480637
4123954
3726631
1794111
3240804

Instancia 3:

TAMAÑO SECUENCIA: 100

SECUENCIA: {120616, 9034, 103362, 83640, 157949, 187070,
29783, 63302, 117203, 56238, 156986, 164808, 83482,
146922, 105441, 13051, 199145, 89817, 84101, 119297,
196259, 7197, 89566, 136981, 68038, 152149, 71872,
4249, 149456, 126268, 175442, 64972, 61069, 177006,
105383, 60183, 179650, 118859, 58298, 42502, 32496,
183860, 26379, 167380, 188997, 173363, 191056, 92895,
28260, 51255, 180762, 570, 76367, 137148, 186374,
119905, 132975, 70473, 93915, 71879, 114305, 159955,

**ANÁLISIS DE ALGORITMOS
1155404-A - Pág: 8
FORMATO DE ANÁLISIS DE
PROBLEMA ALGORÍTMICO**

66055,	66316,	81308,	40319,	190961,	192138,	107618,
168197,	110970,	106127,	163075,	90352,	5718,	85029,
13545,	14978,	126784,	139212,	38020,	76788,	137183,
133193,	182258,	160218,	124915,	155223,	29513,	174437,
95611,	163836,	165573,	66818,	56359,	11703,	99913,
68001,	119797,	57557}				

Resultado de la invocación: 2648926

158005
2301582
1801941
3482325
4230168
563790
1297239
2573250
1130835
3457044
3647007
1798012
3218169
2339769
227044

4571499
1973631
1814127
2621739
4482600
118749
1965457
2992915
1414774
3342475
1508617
63325
3283707
2758969
3917886
1336636
1250515
3961626
2338734
1232766
4038103
2611500
1182867

824536
611922
4141746
484920
3713725
4288413
3865146
4344508
2047458
527058
1022418
4066149
6387
1616484
2996400
4209420
2637235
2904822
1478373
2073402
1508820
2512869
3538414

1363042
1368928
1747344
784440
4342311
4370835
2384556
3736833
2454813
2353968
3605842
1986679
89718
1838055
234693
261930
2770750
3048246
730333
1628028
2997186
2908659
4101273

	3543561 2730322 3412026 556779 3891825 2118088 3623373 3665817 1382070 1133560 204342 2246833 1414029 2634229 1163379
Eficiencia (debe escribir tiempo que demora su solución) para cada instancia. Vea el ejemplo del siguiente ejercicio.	Argumento (Escriba su argumento basado en el punto las instancias de abajo): El método no es el más eficiente ya que realiza múltiples veces el calculo de las sumas, lo que aumenta demasiado el tiempo de procesamiento, para entradas con números pequeños podría considerarse eficiente, sin embargo el tamaño de las entradas y la cantidad de tests es muy grande.

Usé la clase StopWatch.	<p>Instancia 1:</p> <p>TAMAÑO SECUENCIA: 5 SECUENCIA: {86963, 148283, 43246, 51296, 173338}</p> <p>Resultado de la invocación: 41881 microsegundos</p>
	<p>Instancia 2:</p> <p>TAMAÑO SECUENCIA: 50 SECUENCIA: {90360, 190591, 150541, 4200, 71195, 17089, 144093, 158220, 124051, 121644, 38068, 83347, 184841, 49410, 86743, 119813, 192090, 11069, 21753, 57198, 87343, 191400, 133473, 100410, 143995, 83850, 188665, 103866, 35793, 190908, 52469, 167299, 158816, 190748, 23005, 139509, 23948, 148186, 131577, 162649, 41705, 13424, 109657, 105572, 53462, 26158, 183166, 167833, 83316, 147827}</p> <p>Resultado de la invocación: 208271 microsegundos</p>

Instancia 3:

TAMAÑO SECUENCIA: 100

SECUENCIA: {120616, 9034, 103362, 83640, 157949, 187070,
29783, 63302, 117203, 56238, 156986, 164808, 83482,
146922, 105441, 13051, 199145, 89817, 84101, 119297,
196259, 7197, 89566, 136981, 68038, 152149, 71872,
4249, 149456, 126268, 175442, 64972, 61069, 177006,
105383, 60183, 179650, 118859, 58298, 42502, 32496,
183860, 26379, 167380, 188997, 173363, 191056, 92895,
28260, 51255, 180762, 570, 76367, 137148, 186374,
119905, 132975, 70473, 93915, 71879, 114305, 159955,
66055, 66316, 81308, 40319, 190961, 192138, 107618,
168197, 110970, 106127, 163075, 90352, 5718, 85029,
13545, 14978, 126784, 139212, 38020, 76788, 137183,
133193, 182258, 160218, 124915, 155223, 29513, 174437,
95611, 163836, 165573, 66818, 56359, 11703, 99913,
68001, 119797, 57557}

Resultado de la invocación: **442573 microsegundos**

Correctitud	<p>El algoritmo es correcto, ya que da un resultado esperado para todas las entradas posibles</p>
	<p>Instancia 1:</p> <p>TAMAÑO SECUENCIA: 5</p> <p>SECUENCIA: {86963, 148283, 43246, 51296, 173338}</p> <p>Resultado esperado:</p> <p>1890222 3252438 839512 1023225 3864574</p> <p>Resultado del método cándido:</p> <p>1890222 3252438 839512 1023225 3864574</p>
	<p>Instancia 2:</p> <p>TAMAÑO SECUENCIA: 50</p>

ANÁLISIS DE ALGORITMOS
1155404-A - Pág: 16
FORMATO DE ANÁLISIS DE
PROBLEMA ALGORÍTMICO

SECUENCIA: {90360, 190591, 150541, 4200, 71195, 17089,
144093, 158220, 124051, 121644, 38068, 83347, 184841,
49410, 86743, 119813, 192090, 11069, 21753, 57198,
87343, 191400, 133473, 100410, 143995, 83850, 188665,
103866, 35793, 190908, 52469, 167299, 158816, 190748,
23005, 139509, 23948, 148186, 131577, 162649, 41705,
13424, 109657, 105572, 53462, 26158, 183166, 167833,
83316, 147827}

Resultado esperado y el resultado del método cándido coinciden:

1986858
4332664
3309211
62706
1493898
303985
3151653
3489138
2712700
2666520
731269
1794772
4167531

983133
1883386
2634621
4369866
195165
396891
1154769
1900054
4352115
2913873
2254701
3149866
1807599
4277089
2310987
678717
4340907
1046295
3711700
3505647
4336545
419545
3055725

436698
3250135
2876889
3595375
809487
232620
2430721
2342352
1067097
480637
4123954
3726631
1794111
3240804

Instancia 3:

TAMAÑO SECUENCIA: 100

SECUENCIA: {120616, 9034, 103362, 83640, 157949, 187070,
29783, 63302, 117203, 56238, 156986, 164808, 83482,
146922, 105441, 13051, 199145, 89817, 84101, 119297,
196259, 7197, 89566, 136981, 68038, 152149, 71872,
4249, 149456, 126268, 175442, 64972, 61069, 177006,
105383, 60183, 179650, 118859, 58298, 42502, 32496,

183860,	26379,	167380,	188997,	173363,	191056,	92895,
28260,	51255,	180762,	570, 76367,	137148,	186374,	
119905,	132975,	70473,	93915,	71879,	114305,	159955,
66055,	66316,	81308,	40319,	190961,	192138,	107618,
168197,	110970,	106127,	163075,	90352,	5718,	85029,
13545,	14978,	126784,	139212,	38020,	76788,	137183,
133193,	182258,	160218,	124915,	155223,	29513,	174437,
95611,	163836,	165573,	66818,	56359,	11703,	99913,
68001,	119797,	57557}				

Resultado esperado y resultado del método cándido coinciden:

2648926
158005
2301582
1801941
3482325
4230168
563790
1297239
2573250
1130835
3457044
3647007

1798012
3218169
2339769
227044
4571499
1973631
1814127
2621739
4482600
118749
1965457
2992915
1414774
3342475
1508617
63325
3283707
2758969
3917886
1336636
1250515
3961626
2338734

1232766
4038103
2611500
1182867
824536
611922
4141746
484920
3713725
4288413
3865146
4344508
2047458
527058
1022418
4066149
6387
1616484
2996400
4209420
2637235
2904822
1478373

	2073402
	1508820
	2512869
	3538414
	1363042
	1368928
	1747344
	784440
	4342311
	4370835
	2384556
	3736833
	2454813
	2353968
	3605842
	1986679
	89718
	1838055
	234693
	261930
	2770750
	3048246
	730333

	1628028 2997186 2908659 4101273 3543561 2730322 3412026 556779 3891825 2118088 3623373 3665817 1382070 1133560 204342 2246833 1414029 2634229 1163379
Completitud	El método es completo, ya que la lógica usada para hallar la suma de los dígitos, aunque es lento asegura siempre un resultado correcto, si es que este existe

Instancia 1:

TAMAÑO SECUENCIA: 5

SECUENCIA: {86963, 148283, 43246, 51296, 173338}

Resultado esperado:

1890222

3252438

839512

1023225

3864574

Resultado del método cándido:

1890222

3252438

839512

1023225

3864574

Instancia 2:

TAMAÑO SECUENCIA: 50

SECUENCIA: {90360, 190591, 150541, 4200, 71195, 17089,
144093, 158220, 124051, 121644, 38068, 83347, 184841,
49410, 86743, 119813, 192090, 11069, 21753, 57198,

87343,	191400,	133473,	100410,	143995,	83850,	188665,
103866,	35793,	190908,	52469,	167299,	158816,	190748,
23005,	139509,	23948,	148186,	131577,	162649,	41705,
13424,	109657,	105572,	53462,	26158,	183166,	167833,
83316,	147827}					

Resultado esperado y del método cándido coinciden:

1986858
4332664
3309211
62706
1493898
303985
3151653
3489138
2712700
2666520
731269
1794772
4167531
983133
1883386
2634621

4369866
195165
396891
1154769
1900054
4352115
2913873
2254701
3149866
1807599
4277089
2310987
678717
4340907
1046295
3711700
3505647
4336545
419545
3055725
436698
3250135
2876889

3595375
809487
232620
2430721
2342352
1067097
480637
4123954
3726631
1794111
3240804

Instancia 3:

TAMAÑO SECUENCIA: 100

SECUENCIA: {120616, 9034, 103362, 83640, 157949, 187070,
29783, 63302, 117203, 56238, 156986, 164808, 83482,
146922, 105441, 13051, 199145, 89817, 84101, 119297,
196259, 7197, 89566, 136981, 68038, 152149, 71872,
4249, 149456, 126268, 175442, 64972, 61069, 177006,
105383, 60183, 179650, 118859, 58298, 42502, 32496,
183860, 26379, 167380, 188997, 173363, 191056, 92895,
28260, 51255, 180762, 570, 76367, 137148, 186374,
119905, 132975, 70473, 93915, 71879, 114305, 159955,

ANÁLISIS DE ALGORITMOS
1155404-A - Pág: 28
FORMATO DE ANÁLISIS DE
PROBLEMA ALGORÍTMICO

66055,	66316,	81308,	40319,	190961,	192138,	107618,
168197,	110970,	106127,	163075,	90352,	5718,	85029,
13545,	14978,	126784,	139212,	38020,	76788,	137183,
133193,	182258,	160218,	124915,	155223,	29513,	174437,
95611,	163836,	165573,	66818,	56359,	11703,	99913,
68001,	119797,	57557}				

Resultado esperado y resultado del método cándido coinciden:

2648926
158005
2301582
1801941
3482325
4230168
563790
1297239
2573250
1130835
3457044
3647007
1798012
3218169
2339769

227044
4571499
1973631
1814127
2621739
4482600
118749
1965457
2992915
1414774
3342475
1508617
63325
3283707
2758969
3917886
1336636
1250515
3961626
2338734
1232766
4038103
2611500

1182867
824536
611922
4141746
484920
3713725
4288413
3865146
4344508
2047458
527058
1022418
4066149
6387
1616484
2996400
4209420
2637235
2904822
1478373
2073402
1508820
2512869

3538414
1363042
1368928
1747344
784440
4342311
4370835
2384556
3736833
2454813
2353968
3605842
1986679
89718
1838055
234693
261930
2770750
3048246
730333
1628028
2997186
2908659

4101273
3543561
2730322
3412026
556779
3891825
2118088
3623373
3665817
1382070
1133560
204342
2246833
1414029
2634229
1163379

MÉTODO 2

```
public int[] metodoOptimo(int[] casos) {
    int[] result = new int[casos.length];
    for (int i = 0; i < casos.length; i++) {
        result[i] = vec.get(casos[i]);
    }
    return result;
}

private void llenar() {
    for (int i = 0; i <= maxx; i++) {
        this.vec.insertar(0);

    }
    for (int i = 1; i <= maxx; i++) {
        int num = i;
        int suma = 0;
        while (num > 0) {
            suma += num % 10;
            num /= 10;
        }
        suma += this.vec.get(i - 1);
        this.vec.set(i, suma);
    }
}
```

Eficacia	<p>El método es eficaz, puesto a que tiene una solución para todas las entradas válidas ingresadas</p>
	<p>Instancia 1:</p> <p>TAMAÑO SECUENCIA: 5</p> <p>SECUENCIA: {86963, 148283, 43246, 51296, 173338}</p> <p>Resultado de la invocación:</p> <p>1890222 3252438 839512 1023225 3864574</p>
	<p>Instancia 2:</p> <p>TAMAÑO SECUENCIA: 50</p> <p>SECUENCIA: {90360, 190591, 150541, 4200, 71195, 17089, 144093, 158220, 124051, 121644, 38068, 83347, 184841, 49410, 86743, 119813, 192090, 11069, 21753, 57198, 87343, 191400, 133473, 100410, 143995, 83850, 188665, 103866, 35793,</p>

	190908, 52469, 167299, 158816, 190748, 23005, 139509, 23948, 148186, 131577, 162649, 41705, 13424, 109657, 105572, 53462, 26158, 183166, 167833, 83316, 147827}
--	---

Resultado de la invocación:

1986858
4332664
3309211
62706
1493898
303985
3151653
3489138
2712700
2666520
731269
1794772
4167531
983133
1883386
2634621
4369866
195165

	396891
	1154769
	1900054
	4352115
	2913873
	2254701
	3149866
	1807599
	4277089
	2310987
	678717
	4340907
	1046295
	3711700
	3505647
	4336545
	419545
	3055725
	436698
	3250135
	2876889
	3595375
	809487

	<p>232620 2430721 2342352 1067097 480637 4123954 3726631 1794111 3240804</p>
	<p>Instancia 3:</p> <p>TAMAÑO SECUENCIA: 100</p> <p>SECUENCIA: {120616, 9034, 103362, 83640, 157949, 187070, 29783, 63302, 117203, 56238, 156986, 164808, 83482, 146922, 105441, 13051, 199145, 89817, 84101, 119297, 196259, 7197, 89566, 136981, 68038, 152149, 71872, 4249, 149456, 126268, 175442, 64972, 61069, 177006, 105383, 60183, 179650, 118859, 58298, 42502, 32496, 183860, 26379, 167380, 188997, 173363, 191056, 92895, 28260, 51255, 180762, 570, 76367, 137148, 186374, 119905, 132975, 70473, 93915, 71879, 114305, 159955, 66055, 66316, 81308, 40319, 190961, 192138, 107618, 168197, 110970, 106127, 163075, 90352, 5718, 85029, 13545, 14978, 126784, 139212, 38020, 76788,</p>

137183, 133193, 182258, 160218, 124915, 155223, 29513, 174437,
95611, 163836, 165573, 66818, 56359, 11703, 99913,
68001, 119797, 57557}

Resultado de la invocación: 2648926

158005

2301582

1801941

3482325

4230168

563790

1297239

2573250

1130835

3457044

3647007

1798012

3218169

2339769

227044

4571499

1973631

1814127

	2621739
	4482600
	118749
	1965457
	2992915
	1414774
	3342475
	1508617
	63325
	3283707
	2758969
	3917886
	1336636
	1250515
	3961626
	2338734
	1232766
	4038103
	2611500
	1182867
	824536
	611922
	4141746

484920
3713725
4288413
3865146
4344508
2047458
527058
1022418
4066149
6387
1616484
2996400
4209420
2637235
2904822
1478373
2073402
1508820
2512869
3538414
1363042
1368928
1747344

	784440
	4342311
	4370835
	2384556
	3736833
	2454813
	2353968
	3605842
	1986679
	89718
	1838055
	234693
	261930
	2770750
	3048246
	730333
	1628028
	2997186
	2908659
	4101273
	3543561
	2730322
	3412026

	556779 3891825 2118088 3623373 3665817 1382070 1133560 204342 2246833 1414029 2634229 1163379
Eficiencia (debe escribir tiempo que demora su solución) para cada instancia. Vea el ejemplo del siguiente	El método es eficiente ya que calcular una sola vez el método auxiliar reduce notablemente el tiempo de ejecución, ya que no repite operaciones innecesarias. Instancia 1: TAMAÑO SECUENCIA: 5 SECUENCIA: {86963, 148283, 43246, 51296, 173338} Resultado de la invocación: 12 microsegundos

<p><u>ejercicio. Usé la clase StopWatch.</u></p>	<p>Instancia 2:</p> <p>TAMAÑO SECUENCIA: 50</p> <p>SECUENCIA: {90360, 190591, 150541, 4200, 71195, 17089, 144093, 158220, 124051, 121644, 38068, 83347, 184841, 49410, 86743, 119813, 192090, 11069, 21753, 57198, 87343, 191400, 133473, 100410, 143995, 83850, 188665, 103866, 35793, 190908, 52469, 167299, 158816, 190748, 23005, 139509, 23948, 148186, 131577, 162649, 41705, 13424, 109657, 105572, 53462, 26158, 183166, 167833, 83316, 147827}</p> <p>Resultado de la invocación: 24 microsegundos</p>
	<p>Instancia 3:</p> <p>TAMAÑO SECUENCIA: 100</p> <p>SECUENCIA: {120616, 9034, 103362, 83640, 157949, 187070, 29783, 63302, 117203, 56238, 156986, 164808, 83482, 146922, 105441, 13051, 199145, 89817, 84101, 119297, 196259, 7197, 89566, 136981, 68038, 152149, 71872, 4249, 149456, 126268, 175442, 64972, 61069, 177006, 105383, 60183, 179650, 118859, 58298, 42502, 32496, 183860, 26379, 167380,</p>

	<p>188997, 173363, 191056, 92895, 28260, 51255, 180762, 570, 76367, 137148, 186374, 119905, 132975, 70473, 93915, 71879, 114305, 159955, 66055, 66316, 81308, 40319, 190961, 192138, 107618, 168197, 110970, 106127, 163075, 90352, 5718, 85029, 13545, 14978, 126784, 139212, 38020, 76788, 137183, 133193, 182258, 160218, 124915, 155223, 29513, 174437, 95611, 163836, 165573, 66818, 56359, 11703, 99913, 68001, 119797, 57557}</p> <p>Resultado de la invocación: 37 microsegundos</p>
Correctitud	<p>El método es completo, ya que la lógica usada para hallar la suma de los dígitos, es eficiente y asegura siempre un resultado correcto, si es que este existe</p> <p>Instancia 1:</p> <p>TAMAÑO SECUENCIA: 5 SECUENCIA: {86963, 148283, 43246, 51296, 173338}</p> <p>Resultado esperado: 1890222 3252438 839512</p>

1023225
3864574
Resultado del método óptimo:
1890222
3252438
839512
1023225
3864574

Instancia 2:

TAMAÑO SECUENCIA: 50

SECUENCIA: {90360, 190591, 150541, 4200, 71195, 17089, 144093, 158220, 124051, 121644, 38068, 83347, 184841, 49410, 86743, 119813, 192090, 11069, 21753, 57198, 87343, 191400, 133473, 100410, 143995, 83850, 188665, 103866, 35793, 190908, 52469, 167299, 158816, 190748, 23005, 139509, 23948, 148186, 131577, 162649, 41705, 13424, 109657, 105572, 53462, 26158, 183166, 167833, 83316, 147827}

Resultado esperado y el resultado del método óptimo coinciden:

1986858
4332664
3309211

62706
1493898
303985
3151653
3489138
2712700
2666520
731269
1794772
4167531
983133
1883386
2634621
4369866
195165
396891
1154769
1900054
4352115
2913873
2254701
3149866
1807599

4277089
2310987
678717
4340907
1046295
3711700
3505647
4336545
419545
3055725
436698
3250135
2876889
3595375
809487
232620
2430721
2342352
1067097
480637
4123954
3726631
1794111

	3240804
	<p>Instancia 3:</p> <p>TAMAÑO SECUENCIA: 100</p> <p>SECUENCIA: {120616, 9034, 103362, 83640, 157949, 187070, 29783, 63302, 117203, 56238, 156986, 164808, 83482, 146922, 105441, 13051, 199145, 89817, 84101, 119297, 196259, 7197, 89566, 136981, 68038, 152149, 71872, 4249, 149456, 126268, 175442, 64972, 61069, 177006, 105383, 60183, 179650, 118859, 58298, 42502, 32496, 183860, 26379, 167380, 188997, 173363, 191056, 92895, 28260, 51255, 180762, 570, 76367, 137148, 186374, 119905, 132975, 70473, 93915, 71879, 114305, 159955, 66055, 66316, 81308, 40319, 190961, 192138, 107618, 168197, 110970, 106127, 163075, 90352, 5718, 85029, 13545, 14978, 126784, 139212, 38020, 76788, 137183, 133193, 182258, 160218, 124915, 155223, 29513, 174437, 95611, 163836, 165573, 66818, 56359, 11703, 99913, 68001, 119797, 57557}</p> <p>Resultado esperado y resultado del método óptimo coinciden:</p> <p>2648926 158005 2301582</p>

	1801941
	3482325
	4230168
	563790
	1297239
	2573250
	1130835
	3457044
	3647007
	1798012
	3218169
	2339769
	227044
	4571499
	1973631
	1814127
	2621739
	4482600
	118749
	1965457
	2992915
	1414774
	3342475

	1508617
	63325
	3283707
	2758969
	3917886
	1336636
	1250515
	3961626
	2338734
	1232766
	4038103
	2611500
	1182867
	824536
	611922
	4141746
	484920
	3713725
	4288413
	3865146
	4344508
	2047458
	527058

1022418
4066149
6387
1616484
2996400
4209420
2637235
2904822
1478373
2073402
1508820
2512869
3538414
1363042
1368928
1747344
784440
4342311
4370835
2384556
3736833
2454813
2353968

	3605842
	1986679
	89718
	1838055
	234693
	261930
	2770750
	3048246
	730333
	1628028
	2997186
	2908659
	4101273
	3543561
	2730322
	3412026
	556779
	3891825
	2118088
	3623373
	3665817
	1382070
	1133560

	204342 2246833 1414029 2634229 1163379
	El algoritmo es completo, ya que da un resultado esperado para todas las entradas posibles
Completitud	Instancia 1: TAMAÑO SECUENCIA: 5 SECUENCIA: {86963, 148283, 43246, 51296, 173338} Resultado esperado: 1890222 3252438 839512 1023225 3864574 Resultado del método óptimo: 1890222 3252438 839512 1023225

	3864574
	<p>Instancia 2:</p> <p>TAMAÑO SECUENCIA: 50</p> <p>SECUENCIA: {90360, 190591, 150541, 4200, 71195, 17089, 144093, 158220, 124051, 121644, 38068, 83347, 184841, 49410, 86743, 119813, 192090, 11069, 21753, 57198, 87343, 191400, 133473, 100410, 143995, 83850, 188665, 103866, 35793, 190908, 52469, 167299, 158816, 190748, 23005, 139509, 23948, 148186, 131577, 162649, 41705, 13424, 109657, 105572, 53462, 26158, 183166, 167833, 83316, 147827}</p> <p>Resultado esperado y el resultado del método óptimo coinciden:</p> <p>1986858 4332664 3309211 62706 1493898 303985 3151653 3489138 2712700 2666520</p>

731269
1794772
4167531
983133
1883386
2634621
4369866
195165
396891
1154769
1900054
4352115
2913873
2254701
3149866
1807599
4277089
2310987
678717
4340907
1046295
3711700
3505647

	4336545 419545 3055725 436698 3250135 2876889 3595375 809487 232620 2430721 2342352 1067097 480637 4123954 3726631 1794111 3240804
	<p>Instancia 3:</p> <p>TAMAÑO SECUENCIA: 100</p> <p>SECUENCIA: {120616, 9034, 103362, 83640, 157949, 187070, 29783, 63302, 117203, 56238, 156986, 164808, 83482, 146922, 105441, 13051, 199145, 89817, 84101, 119297, 196259, 7197,</p>

	<p>89566, 136981, 68038, 152149, 71872, 4249, 149456, 126268, 175442, 64972, 61069, 177006, 105383, 60183, 179650, 118859, 58298, 42502, 32496, 183860, 26379, 167380, 188997, 173363, 191056, 92895, 28260, 51255, 180762, 570, 76367, 137148, 186374, 119905, 132975, 70473, 93915, 71879, 114305, 159955, 66055, 66316, 81308, 40319, 190961, 192138, 107618, 168197, 110970, 106127, 163075, 90352, 5718, 85029, 13545, 14978, 126784, 139212, 38020, 76788, 137183, 133193, 182258, 160218, 124915, 155223, 29513, 174437, 95611, 163836, 165573, 66818, 56359, 11703, 99913, 68001, 119797, 57557}</p>
--	--

Resultado esperado y resultado del método óptimo coinciden:

2648926

158005

2301582

1801941

3482325

4230168

563790

1297239

2573250

1130835

3457044
3647007
1798012
3218169
2339769
227044
4571499
1973631
1814127
2621739
4482600
118749
1965457
2992915
1414774
3342475
1508617
63325
3283707
2758969
3917886
1336636
1250515

	3961626
	2338734
	1232766
	4038103
	2611500
	1182867
	824536
	611922
	4141746
	484920
	3713725
	4288413
	3865146
	4344508
	2047458
	527058
	1022418
	4066149
	6387
	1616484
	2996400
	4209420
	2637235

2904822
1478373
2073402
1508820
2512869
3538414
1363042
1368928
1747344
784440
4342311
4370835
2384556
3736833
2454813
2353968
3605842
1986679
89718
1838055
234693
261930
2770750

	3048246
	730333
	1628028
	2997186
	2908659
	4101273
	3543561
	2730322
	3412026
	556779
	3891825
	2118088
	3623373
	3665817
	1382070
	1133560
	204342
	2246833
	1414029
	2634229
	1163379