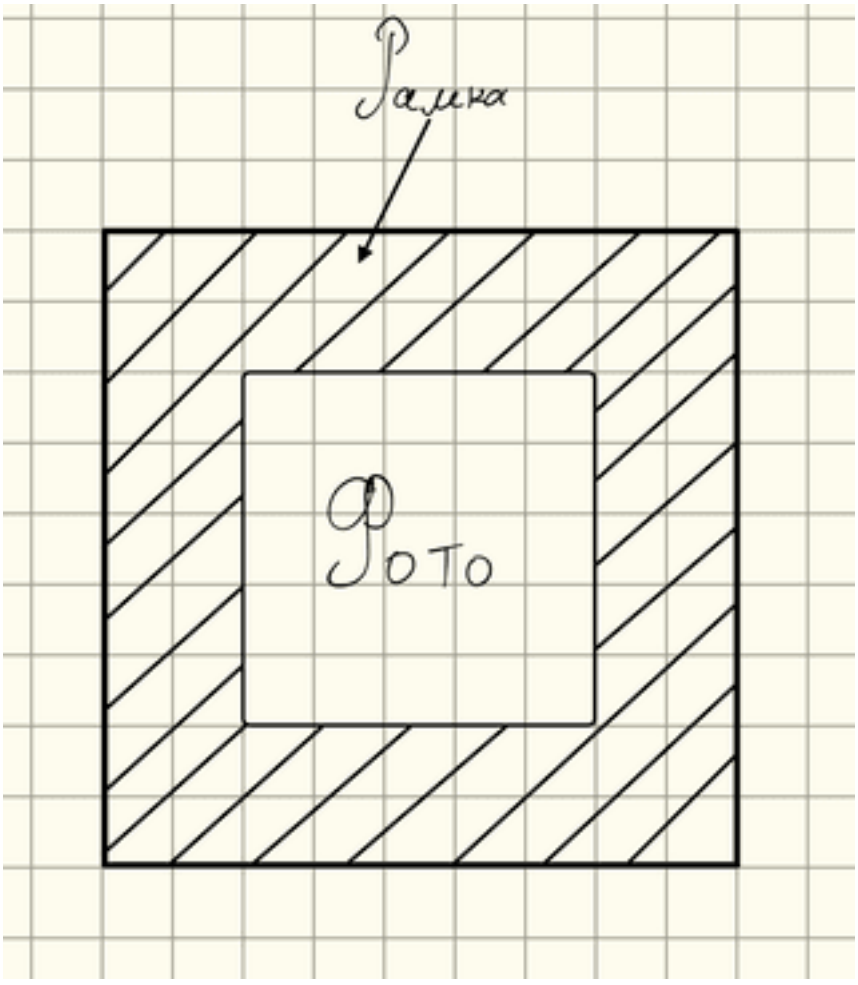
Задача A. Фотографія

Обмеження використання часу:	1 секунда
Обмеження використання пам'яті:	256 мегабайтів

Зробивши фотографію, Василько захотів поставити її в рамку. Фотографія Василька має розмір $n \times n$, рамка повинна мати товщину t.

Порахуйте, яку площу буде мати рамка для фотографії Василька.

Наприклад, фотографія розміру 5 з шириною рамки 2 виглядає так:



Формат вхідних даних

Перший рядок містить два цілі числа n, t ($3 \leqslant n \leqslant 10^9$; $1 \leqslant t$; $2t < n$) — розмір фотографії, товщина рамки.

Формат вихідних даних

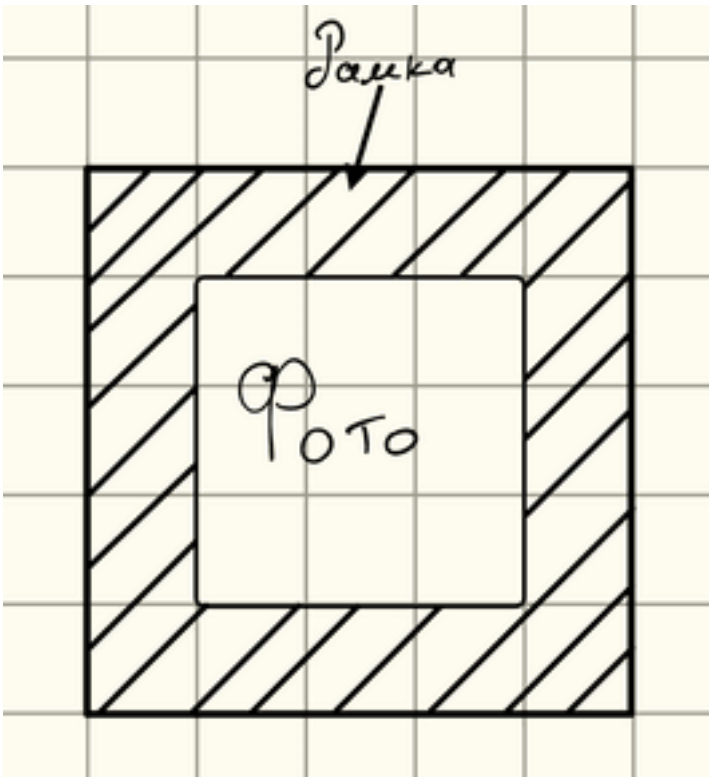
Одне ціле число, площа рамки для фотографії Василька.

Приклади

стандартний ввід	стандартний вивід
3 1	16
5 2	56

Примітка

Так виглядає рамка для першого прикладу:



Задача B. Футбол

Обмеження використання часу:	1 секунда
Обмеження використання пам'яті:	256 мегабайтів

Одного разу Василько, граючи в футбол, не зміг забити гол.

Повернувшись до дому, Василько згадав, що футбольне поле має вигляд координатної площини.

Також Василько знаходиться в позиції x_v, y_v на цьому полі. Ворота можна представити за допомогою трьох цілих чисел x, y_1, y_2, і кожна пара x, y, де $y_1 \leqslant y \leqslant y_2$, належить до воріт.

Припустимо, що Василько вирішив забити м'яч у точку з координатами x_f, y_f, де $x_f = x$ і $y_1 \leqslant y \leqslant y_2$. Тоді, щоб м'яч потрапив у цю точку, йому потрібна сила удару, яка рівна $p = |x_f - x_v| + |y_f - y_v|$.

Допоможіть Васильку порахувати, з якою мінімальною силою він повинен був вдарити, щоб забити гол.

Формат вхідних даних

Перший рядок містить п'ять цілих чисел x, y_1, y_2, x_v, y_v ($1 \leqslant x, y_1, y_2, x_v, y_v \leqslant 10^9, y_1 \leqslant y_2$) — позиції воріт та позиція Василька. Гарантовано, що $x \neq x_v$.

Формат вихідних даних

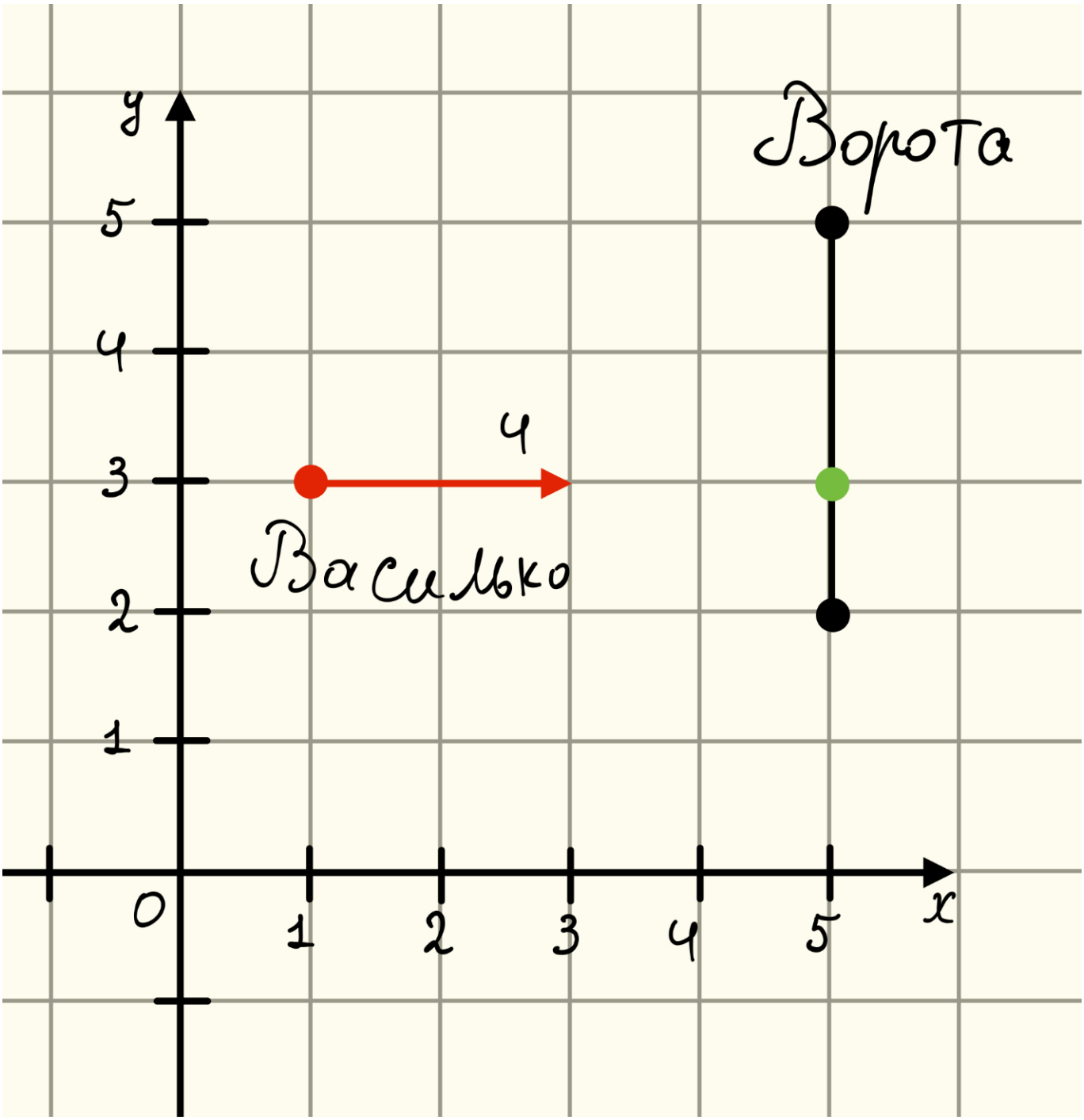
Виведіть одне ціле число, мінімальну силу удару Василька.

Приклад

стандартний ввід	стандартний вивід
5 2 5 1 3	4

Примітка

В прикладі так виглядатиме удар Василька:


y
5
4
3
2
1
0
1
2
3
4
5
x
Ворота
Ч
Василько

Задача C. Пікнік

Обмеження використання часу:	1 секунда
Обмеження використання пам'яті:	256 мегабайтів

Василько має пиріг, розрізаний на n шматочків. Вага кожного шматочка становить $a_1, a_2, \ldots, a_n$. Він хоче поділити всі шматочки між собою та k друзями так, щоб кожен отримав порцію однакової ваги.

Пиріг потрібно ділити за такими правилами:

- Кожна людина отримує кілька шматочків (можливо, один), що йдуть у масиві один за одним (неперервний блок).
- Сумарна вага шматочків у кожній порції має бути однаковою.
- Всього пиріг ділиться на $k + 1$ рівну частину (тобто сам Василько та k друзів). Завдання задачі полягає в тому, щоб знайти максимальне число k (кількість друзів), при якому такий поділ можливий. Зверніть увагу, що кожен шматок має бути у якісь порції.

Допоможіть Васильку порахувати, яку максимальну кількість друзів він може запросити на пікнік.

Формат вхідних даних

Перший рядок містить одне число n ($1 \leqslant n \leqslant 5000$) — кількість кусочків пирога.

Другий рядок містить n цілих чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant 10^4$) — вага кожного кусочка пирога.

Формат вихідних даних

Одне ціле число, максимальну кількість друзів, котрих може запросити Василько.

Приклади

стандартний ввід	стандартний вивід
11 1 3 4 2 2 3 1 4 1 1 2	5
5 2 3 2 1 3	0

Примітка

У першому прикладі він може розділити пиріг на 6 частин, де вага кожної частини буде 4. Це можна зробити, наприклад, так: $1 + 3$, 4, $2 + 2$, $3 + 1$, 4, $1 + 1 + 2$. Оскільки буде всього 6 частин, то він може запросити лише 5 друзів.

У другому прикладі він може лише розділити пиріг на 1 частину (весь пиріг). У такому випадку він не може запросити друзів.

Задача D. Комфорт Василька

Обмеження використання часу:	1 секунда
Обмеження використання пам'яті:	256 мегабайтів

Недавно Василько вирішив, що він буде записувати температуру повітря кожну хвилину. І до тепер в нього назбиралося n записів $a_1, a_2, \ldots, a_n$.

Проміжок часу можна назвати комфортним для Василька, тоді коли різниця між максимальною і мінімальною температурою була не більшою за k.

Знайдіть найдовший період часу з записів Василька, коли йому було комфортно.

Формат вхідних даних

Перший рядок містить два цілі числа n та k ($1 \leqslant n \leqslant 2 \cdot 10^5, 0 \leqslant k \leqslant 10^9$) — кількість записів Василька та максимальна різниця температур.

Другий рядок містить n цілих чисел $a_1, a_2, \ldots, a_n$ ($0 \leqslant a_i \leqslant 10^9$) — значення кожного з записів Василька.

Формат вихідних даних

Одне ціле число, довжина найдовшого періоду часу, коли Васильку було комфортно.

Приклад

стандартний ввід	стандартний вивід
10 3 1 3 2 5 2 7 1 2 4 3	4

Примітка

Проміжок від 7 хвилини до 10 хвилини ($[1, 2, 4, 3]$) є комфортним для Василька. Усі періоди часу довжини 5 містять пари 7 і 1, 7 і 2 або 5 і 1, тому вони не будуть комфортними.

Задача E. Гра

Обмеження використання часу:	1 секунда
Обмеження використання пам'яті:	256 мегабайтів

Недавно граючись зі своїми друзями, Василько придумав нову гру. Сьогодні він вирішив поділитися нею з тобою.

В цій грі потрібно збудувати масив a довжини n, у якого такі властивості:

- кожне число має бути від $-m$ до m;
- серед усіх можливих підвідрізків масиву є лише один, у якому сума чисел максимальна, і це підвідзок з l до r включно.

Вам потрібно побудувати такий масив, проте серед усіх таких можливих масивів вам потрібно побудувати саме той, у якому сума невід'ємних чисел максимальна.

Формат вхідних даних

Перший рядок містить чотири цілі числа n, m, l, r ($1 \leqslant n \leqslant 10^5$, $1 \leqslant m \leqslant 10^9$, $1 \leqslant l \leqslant r \leqslant n$) — довжина масиву, обмеження на значення масиву, краї відрізка.

Формат вихідних даних

В одному рядку потрібно виписати побудований масив a довжини n.

Якщо відповідей декілька, то ви можете вивести будь-який масив.

Приклад

стандартний ввід	стандартний вивід
5 3 2 3	-1 3 3 -3 2

Примітка

Сума на кожному з відрізків з прикладу:

- В відрізку від 1 до 1 сума -1.
- В відрізку від 1 до 2 сума 2.
- В відрізку від 1 до 3 сума 5.
- В відрізку від 1 до 4 сума 2.
- В відрізку від 1 до 5 сума 4.
- В відрізку від 2 до 2 сума 3.
- В відрізку від 2 до 3 сума 6.
- В відрізку від 2 до 4 сума 3.
- В відрізку від 2 до 5 сума 5.
- В відрізку від 3 до 3 сума 3.

- В відрізку від 3 до 4 сума 0.
- В відрізку від 3 до 5 сума 2.
- В відрізку від 4 до 4 сума -3.
- В відрізку від 4 до 5 сума -1.
- В відрізку від 5 до 5 сума 2.

тому на відрізку від 2 до 3 буде максимальна сума, і він єдиний з такою сумою.

Задача A. Фотографія

Автор: Андрій Холод
Роздробник: Павло Цікалишин
Розбір підготовано: Костянтин Денисов

Фотографія має розмір $n \times n$. Рамка товщини t додає смугу шириною t з кожного боку, тому зовнішній розмір усієї конструкції дорівнює $(n + 2t) \times (n + 2t)$. Площа рамки дорівнює різниці площ зовнішнього квадрата та фотографії.

Площа фотографії: $S_{\text{in}} = n^2$.
Площа зовнішнього квадрата: $S_{\text{out}} = (n + 2t)^2$.
Отже, площа рамки: $S = S_{\text{out}} - S_{\text{in}} = (n + 2t)^2 - n^2$.

Спростимо вираз: $(n + 2t)^2 - n^2 = n^2 + 4nt + 4t^2 - n^2 = 4t(n + t)$.

Тому відповідь: $S = 4t(n + t)$.

Складність. Час: $O(1)$, пам'ять: $O(1)$.

Зауваження. За $n \leqslant 10^9$ значення площ може бути порядку 10^{18}, тому слід використовувати 64-бітний цілий тип (наприклад, `long long`).

Задача B. Футбол

Автор: Андрій Холод
Роздробник: Павло Цікалишин
Розбір підготовано: Костянтин Денисов

Щоб забити гол, достатньо влучити в будь-яку точку воріт, тобто мінімізувати $|x - x_v| + |y - y_v|$ по всіх $y \in [y_1, y_2]$.

Частина $|x - x_v|$ не залежить від y, тому мінімізувати треба лише $|y - y_v|$ на відрізку $[y_1, y_2]$. Мінімальна відстань від точки y_v до відрізка $[y_1, y_2]$ дорівнює:

- 0, якщо $y_1 \leqslant y_v \leqslant y_2$,
- $y_1 - y_v$, якщо $y_v < y_1$,
- $y_v - y_2$, якщо $y_v > y_2$.

Це можна записати одним рядком як $d_y = \max(0,\ y_1 - y_v,\ y_v - y_2)$.

Тоді відповідь: $p_{\min} = |x - x_v| + d_y$.

Складність. Час: $O(1)$, пам'ять: $O(1)$.

Задача C. Пікнік

Автор: Андрій Холод
Роздробник: Павло Цікалишин
Розбір підготовано: Костянтин Денисов

Потрібно розбити масив $a_1, \ldots, a_n$ на $m = k + 1$ **послідовних** частин так, щоб сума в кожній частині була однакова, і використати всі шматки. Нехай $S = \sum_{i=1}^{n} a_i$. Тоді кожна частина має вагу $T = S/m$, отже обов'язково $m \mid S$.

Через те, що $n \leqslant 5000$, можна зробити дуже просту перевірку: перебрати всі можливі m від найбільшого до 1 і перевірити, чи можна нарізати на m частин сумою T жадібно зліва направо. Оскільки ми знаємо наперед суму в кожній частині, то далі перший відрізок визначається однозначно, за ним другий і так далі.

Перевірка для фіксованого m. Обчислюємо $T = S/m$ і йдемо по масиву, накопичуючи cur та підраховуючи кількість друзів cnt:

- $cur := cur + a_i$.
- якщо $cur = T$, то "закриваємо"частину: $cnt := cnt + 1$, $cur := 0$;
- якщо $cur > T$, то таке m неможливе (бо всі $a_i \geqslant 1$ і ми вже перебрали суму).

В кінці потрібно, щоб $cur = 0$ і $cnt = m$.

Складність. Перебираємо m не більше ніж n разів, кожна перевірка за $O(n)$, отже загалом $O(n^2)$. За $n \leqslant 5000$ це проходить. Пам'ять $O(n)$.

Задача D. Комфорт Василька

Автор: Андрій Холод
Роздробник: Павло Цікалишин
Розбір підготовано: Костянтин Денисов

Потрібно знайти найдовший підвідрізок масиву $a_1, \ldots, a_n$, для якого $\max(a_l, \ldots, a_r) - \min(a_l, \ldots, a_r) \leqslant k$. Це класична задача на два вказівники з підтримкою поточного мінімуму і максимуму в вікні.

Якщо фіксувати праву межу r і збільшувати l, то max не зростає, а min не спадає, отже різниця $\max - \min$ не збільшується. Тому можна рухати два вказівники: розширювати вікно вправо, а коли умова порушується — зсувати ліву межу, поки знову не стане добре.

Проблема: потрібно швидко знати min і max у поточному вікні $[l, r]$. Для цього підтримуємо `std::multiset` на елементах поточного вікна.

Алгоритм завжди підтримує інваріант: після внутрішнього циклу вікно $[l, r]$ є комфортним. Для кожного r ми мінімізуємо l (зсуваємо його лише поки треба), тому довжина $r - l + 1$ є максимальною серед усіх комфортних підвідрізків, що закінчуються в r. Максимум по всіх r дає відповідь задачі.

Складність. Кожен індекс додається в множину рівно один раз і видаляється не більше одного разу, тому загальний час $O(n \log n)$, пам'ять $O(n)$.

Задача E. Гра

Автор: Костянтин Денисов
Роздробник: Павло Цікалишин
Розбір підготовано: Костянтин Денисов

Очевидно, що на відрізку $[l, r]$ треба ставити якнайбільші числа, тобто $a_i = m$ для всіх $l \leqslant i \leqslant r$. Тоді цей відрізок має найбільший потенціал, і далі наша задача — зробити так, щоб жоден інший підвідрізок не міг з ним зрівнятися за сумою, та ще й серед усіх таких масивів максимізувати суму невід'ємних елементів.

Окремий випадок $l = r$. Тут максимальна сума має бути рівно m і досягатися **лише** на позиції l. Тому поза l не можна ставити m, інакше одиночний підвідрізок з цієї позиції теж матиме суму m. Отже поза l найкраще ставити $m - 1$. Але треба ще зіпсувати всі підвідрізки, які включають l і ще щось: для цього достатньо розставити $-m$ на кожній другій позиції, починаючи від сусідів $l - 1$ та $l + 1$. Тоді будь-яке розширення відрізка за межі $[l, l]$ неминуче підхопить хоча б один $-m$, і сума стане строго меншою за m.

Випадок $l < r$.

Поза відрізком $[l, r]$ оптимальна конфігурація має максимально багато m, але такі m доводиться ставити через один, чергуючи з $-m$; інакше з'являється занадто великий підвідрізок поза $[l, r]$ або підвідрізок, що "приклеюється"до $[l, r]$.

Тому для **префікса** довжини $s = l - 1$ оптимально ставити (зліва направо):

- якщо s **парна**, то $m, -m, m, -m, \ldots, m, -m, m - 1, -m$; доводиться ставити одну $m - 1$, щоб сума елементів була від'ємна, бо інакше буде декілька елементів з однаковою максимальною сумою.
- якщо s **непарна**, то $m, -m, m, -m, \ldots, m, -m, -m$.

Для **суфікса** довжини $t = n - r$ робимо аналогічну (дзеркальну) побудову за парністю t.

У будь-якому допустимому масиві поза $[l, r]$ не можна ставити багато великих додатних підряд: кожен додатний m змушений мати поруч достатню "компенсацію"$-m$, інакше з'являється підвідрізок з занадто великою сумою, який конкурує з $[l, r]$. Отже m можна ставити не частіше, ніж через один, тобто шаблон $m, -m, m, -m, \ldots$ дає максимальну кількість m. У парному випадку рівно один раз доводиться замінити m на $m - 1$, щоб уникнути рівності максимумів.

Складність. Побудова займає $O(n)$ часу і $O(1)$ додаткової пам'яті (окрім виводу).