# Project Euler #222: Sphere Packing | HackerRank challenges

What is the length of the shortest pipe, of internal radius $R$ mm, that can fully contain $n$ balls of radii $r_i$ where $0 \le i \le n - 1$?

Give your answer in micrometers ($10^{-6}$ m) rounded to the nearest integer.

**Input Format**

The first line of each test file contains two-separated integers $R$ and $n$.
The next line contains $n$ space-separated integers $r_0 , \ldots , r_{n-1}$ .

**Constraints**

- $2 \le R \le 10^6$
- $1 \le n \le 4 \cdot 10^5$
- $7 \cdot R/13 \le r_i \le R$
- $r_i$ are pairwise distinct

**Output Format**

Print your answer in one line.

**Sample Input 0**

2 1
2

**Sample Output 0**

4000

**Sample Input 1**

5 2
3 4

**Sample Output 1**

13325

**Sample Input 2**

100 5

61 62 63 64 65

**Sample Output 2**

530707

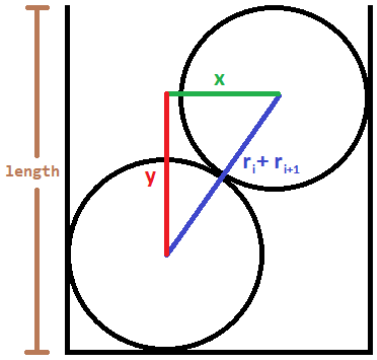
# Solution

## Geometric analysis

Since the diameter of each ball is greater than the radius of the pipe, the problem is simplified to circle packing in a rectangle (2D). The key element to the mathematical part of the problem is to calculate the pipe-wise distance of the centers of any 2 consecutive circles, which is the red y distance at the picture. So, using the Pythagorean theorem:

$$y_i = \sqrt{(r_i + r_{i+1})^2 - x_i^2} = \sqrt{(r_i + r_{i+1})^2 - [R - (r_i + r_{i+1})]^2}$$

$$\boxed{y_i = 2\sqrt{R\,(r_i + r_{i+1} - R)}}$$




And, finally the length of the pipe will be the summation of all the y distances, plus the radii of the first and the last circles:

$$\boxed{l = r_0 + r_n + \sum_{i=1}^{n} \left(2\sqrt{R\,(r_{i-1} + r_i - R)}\right)}$$

In order to understand the placement sequence, it's enough to demonstrate 2 examples: 3 and 4 circles.

Example 1:

- R = 100mm
- n = 3
- sphere radii: 65mm, 64mm, 63mm

$$l = r_0 + r_3 + 2\sqrt{R\,(r_0 + r_1 - R)} + 2\sqrt{R\,(r_1 + r_2 - R)}$$

The best arrangement is to place the 2 bigger spheres at the edges. That is:

$r_0$ = 65mm, $r_1$ = 63mm, $r_2$ = 64mm

You can do the math to find out yourself, but a simple observation is that first and last spheres take part in only 1 square root, while all the others can be found in 2. So, you choose the bigger ones to be first and last.

Example 2:

- R = 100mm
- n = 4
- sphere radii: 65mm, 64mm, 63mm, 62mm

$$l = r_0 + r_3 + 2\sqrt{R\,(r_0 + r_1 - R)} + 2\sqrt{R\,(r_1 + r_2 - R)} + 2\sqrt{R\,(r_2 + r_3 - R)}$$

Again, first place the 2 bigger spheres at the edges. The question is whether you put the 3rd bigger (63mm) after the 1st (0th) or next to the last. Again, we will do the math:

1. Case $r_0$ = 65mm, $r_1$ = 62mm, $r_2$ = 63mm, $r_3$ = 64mm

$$2\sqrt{R}\left(\sqrt{(r_0 + r_1 - R)} + \sqrt{(r_1 + r_2 - R)} + \sqrt{(r_2 + r_3 - R)}\right)$$

It all comes down to the comparison of the summation of these square roots:

$$\sqrt{(r_0 + r_1 - R)} + \sqrt{(r_1 + r_2 - R)} + \sqrt{(r_2 + r_3 - R)} = \sqrt{27} + \sqrt{25} + \sqrt{27}$$

2. Case $r_0$ = 65mm, $r_1$ = 63mm, $r_2$ = 62mm, $r_3$ = 64mm

$$\sqrt{(r_0 + r_1 - R)} + \sqrt{(r_1 + r_2 - R)} + \sqrt{(r_2 + r_3 - R)} = \sqrt{28} + \sqrt{25} + \sqrt{26}$$

Finally, $\sqrt{28} + \sqrt{26} < \sqrt{27} + \sqrt{27}$. Consequently, the best fit is an alternately arrangement.

## Code strategy

First, we have to construct the structure that contains the radii with the placement sequence. So, the input radii are sorted in a vector. Then, this vector is separated in two, in an alternately manner. And, finally, the second is added to the 1st, backwards.

Eventually, there is a recursive summation of the y distances.

Notice (requirements to pass all the tests):

1. Print via the arithmetic format and not the default scientific. In C++:
   #include <iomanip>
   std::cout << std::fixed << std::setprecision(0) << length;
2. In order to include the errors of the naive summation of floating point numbers - square roots (especially with an aggressive compiler optimizer), the Kahan summation method is implemented. Testing with few summations, the errors are limited at the order of $10^{-18}$ and tend to oscillate around zero (in a random walk manner), but after around $10^5$ summations, error indeed sum up to be significant.