Interview Question

- Find all possible values for a, b, c and d in the following equation
  - $a^3 + b^3 = c^3 + d^3$
  - Given: 1 <= a, b, c, d <= 1000
  - Example:
    - a, b, c, d values , say, 1, 1, 1, 1 will satisfy the given equation
    - a, b, c, d values , say, 10, 10, 10, 10 will satisfy the given equation

Find all possible values for a, b, c and d in the following equation:

$a^3 + b^3 = c^3 + d^3$ Given: 1 <= a, b, c, d <= 1000

```
N = 1000
for a  from 1 to N
    for b  from 1 to N
        for c  from 1 to N
            for d  from 1 to N
                if ( $a^3 + b^3 == c^3 + d^3$ )
                        print a, b, c d;
```

What is the time complexity of this solution?

What about space complexity?

If u look closely, u will see that the innermost for loop is not really needed… Why?

Innermost for loop in the previous slide is not needed because we can compute the required value of d at that point.

So, removing the innermost for loop, we get:

```
N = 1000
for a  from 1 to N
    for b  from 1 to N
        for c  from 1 to N
            d = $(a^3 + b^3 - c^3)^{1/3}$        ← Just compute the value of d here, round that to int
            if ( $a^3 + b^3 == c^3 + d^3$ )  ← Check if the rounded value of d works.
                    print a, b, c d;   ← If value of d worked, then we print
```

What is the time complexity now ?

What about space complexity?

We can still make improvements to the solution.

Notice that the (c, d) combinations that fit into the required equation may be computed multiple times.

For example:

- if c is, say, 5, and d is, say, 10, then $c^3 + d^3$ is 125 + 1000 = 1125
- Now, when c is 10 and d is 5, we will have the same sum: $c^3 + d^3$ is 1000 + 125 = 1125
- There could be many such pairs.

```
N = 1000
for c  from 1 to N
    for d  from 1 to N
        sum = $c^3 + d^3$
        Store sum and the pair (c, d). ← Store key, value pair, where key is sum, value is the pair (c,d)
```

Note that a given sum may have multiple (c, d) pairs stored against it.

And these (c, d) pairs can also be thought of as (a, b) pairs, since range of a,b,c,d is the same

1 <= a, b, c, d <= 1000

So, we now go through our stored sum entries.

```
For each sum
        Get the list of (c, d) pairs we had stored against that sum
        for each pair p1 in this list
            for each pair p2 in this list
                print p1, p2   ← This is a total of 4 numbers, since each of p1 and p2 is a pair
```

For example, a sum of 1729 will have these pairs:

1 12

9 10

10 9

12 1

(1,12,1,12), (1,12, 9, 10), (1,12, 10, 9), (1,12,12,1)

(9, 10, 1, 12) …

What is the time complexity now ?