

Homework of algorithm analysis

软工一班 成子谦 201730681303

Chapter. 1

We can solve this problem by binary search.

Source code:

```
#include <iostream>
#include <climits>
#include <cmath>

using namespace std;

int main() {
    int l = 0, r = INT_MAX, ans;
    while (l <= r) {
        int mid = l + r >> 1;
        if (1000 * mid * mid <= pow(2, mid)) r = mid - 1;
        else l = mid + 1;
    }
    cout << l << endl;
    return 0;
}
```

And the result is 19, which means $1000 \cdot N^2$ is smaller than 2^N when N is equal or larger than 19.

Chapter. 2

The time complexity of the first code is $O(N \log N)$, the second one is $O(N^3)$.

Chapter. 3

We can solve this problem by enumerate all possible positions of numbers.

Source code:

```
#include <cstdio>
#include <vector>
#include <algorithm>
#include <numeric>

using namespace std;

vector<int>a(10);

int check(int &x, int &y, int &z) {
```

```
    for (int i = 0; i < 3; i++) x = x * 10 + a[i];
    for (int i = 3; i < 5; i++) y = y * 10 + a[i];
    for (int i = 5; i < 10; i++) z = z * 10 + a[i];
    return x * y == z ? 1 : 0;
}

int main() {
    iota(a.begin(), a.end(), 0); // init vector into 0, 1, 2, ..., 9
    do {
        int x = 0, y = 0, z = 0;
        check(x, y, z) ? printf("x = %d, y = %d, z = %d\n", x, y, z) : 0;
    } while (next_permutation(a.begin(), a.end()));
    return 0;
}
```

And the result is:

```
x = 138, y = 42, z = 5796
x = 157, y = 28, z = 4396
x = 159, y = 48, z = 7632
x = 186, y = 39, z = 7254
x = 198, y = 27, z = 5346
x = 297, y = 18, z = 5346
x = 297, y = 54, z = 16038
x = 345, y = 78, z = 26910
x = 367, y = 52, z = 19084
x = 396, y = 45, z = 17820
x = 402, y = 39, z = 15678
x = 483, y = 12, z = 5796
x = 495, y = 36, z = 17820
x = 594, y = 27, z = 16038
x = 715, y = 46, z = 32890
x = 927, y = 63, z = 58401
```