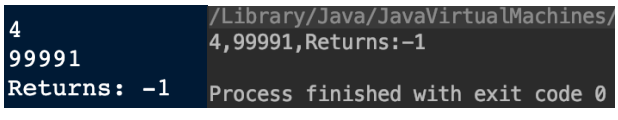
The type of Graph Theory contains a lot of problems. This week I still practice this type of the problem. The problem is DivisorInc. This is an interesting problem.

First of all, what I mean by this is that for a number K that's greater than or equal to 4, you can add one of its factors (1, except for K) to get another number. Given N and M (N less than or equal to M), find the minimum number of operations from N to M. Using factors of N (not including 1, N) For eg: N=4, M=24: 4 -> 6(4+2) -> 8(6+2) -> 12(8+4) ->18(12+6) ->24. Instead of having a graph, I must build the graph as I go.

So, I should define that int MAX = 100001, d [] = new int[MAX]; Then, I use the For loop and if statement.
The core part of the code is like this:

```
for( int i = 0; i < MAX; i++ ) d[i] = MAX;
d[N] = 0;
for( int i = N; i <= M; i++ ) if( d[i] < MAX )
    for( int j = 2; j * j <= i; j++ )
        if( i % j == 0 )
{
    if( i + j < MAX && d[i+j] > d[i] + 1 ) d[i+j] = d[i] + 1;
    if( i + i/j < MAX && d[i+i/j] > d[i] + 1 ) d[i+i/j] = d[i] + 1;
}
return d[M] < MAX ? d[M] : -1;
```

Class: DivisorInc
Test passed.
Since I can't upload it to web submission, the example is shown below and the result looks like this after I run it.



The challenge of this problem is to make sure the thinking of this problem in my mind is clear so that I can solve it easier. To solve the graph problem, the normal method is to use BFS though in a slightly different way.
https://community.topcoder.com/stat?c=problem_statement&pm=6186