Here is an implemented algorithm:

```
#include <iostream>
int count;

void do_something_costly() {
    count++;
}

void f(int n) {
    for (int i=0; i<n*100; i++) do_something_costly();
    for (int i=0; i<n; i++)
        for (int j=0; j<=i; j++)
            do_something_costly();
}

int main() {
    int sizes[] = {1, 10, 20, 30, 40, 50, 60, 70, 80, 90,
                         100, 200, 300, 400, 500, 600, 700, 800, 900,
                         1000, 2000, 3000, 4000, 5000, 6000, 7000,
                         8000, 9000, -1};
    int i=0;
    while (true) {
        int n = sizes[i++];
        if (n == -1) break;
        count = 0;
        f(n);
        std::cout << n << "," << count << "\n";
    }
}
```

Run this program on the console, and redirect the output to a file:
`example.exe > cost.csv`
note the `.csv` ending. MS Excel recognizes these files. The output is:

```
1,101
10,1055
20,2210
30,3465
40,4820
50,6275
60,7830
70,9485
80,11240
90,13095
100,15050
200,40100
300,75150
400,120200
500,175250
600,240300
700,315350
800,400400
900,495450
1000,600500
```

```
2000,2201000
3000,4801500
4000,8402000
5000,13002500
6000,18603000
7000,25203500
8000,32804000
9000,41404500
```

Let's double-click on this file to open it with Excel:

	A	B
1	1	101
2	10	1055
3	20	2210
4	30	3465
5	40	4820
6	50	6275
7	60	7830
8	70	9485
9	80	11240
10	90	13095
11	100	15050
12	200	40100
13	300	75150
14	400	120200
15	500	175250
16	600	240300
17	700	315350
18	800	400400
19	900	495450
20	1000	600500
21	2000	2201000
22	3000	4801500
23	4000	8402000
24	5000	13002500
25	6000	18603000
26	7000	25203500
27	8000	32804000
28	9000	41404500

Column A has n, and B has $C(n)$, the cost of running the function with input size n.

Now, we suspect this is a quadratic function:

$C(n) = An^2$

If we take logs of both sides, we get:

$\log(C(n)) = \log(A) + 2\log(n)$

Thus we expect that a plot of $\log(C(n))$ versus $\log(n)$ should be a line with slope 2. Let's verify this. First, compute the logs in columns C and D:

Move the cursor to cell C1, and type `=log(A1)` . Then go to cell D1, and type `=log(B1)`.

Then, select cells C1 and D1, and copy them into rows 2 through 28:

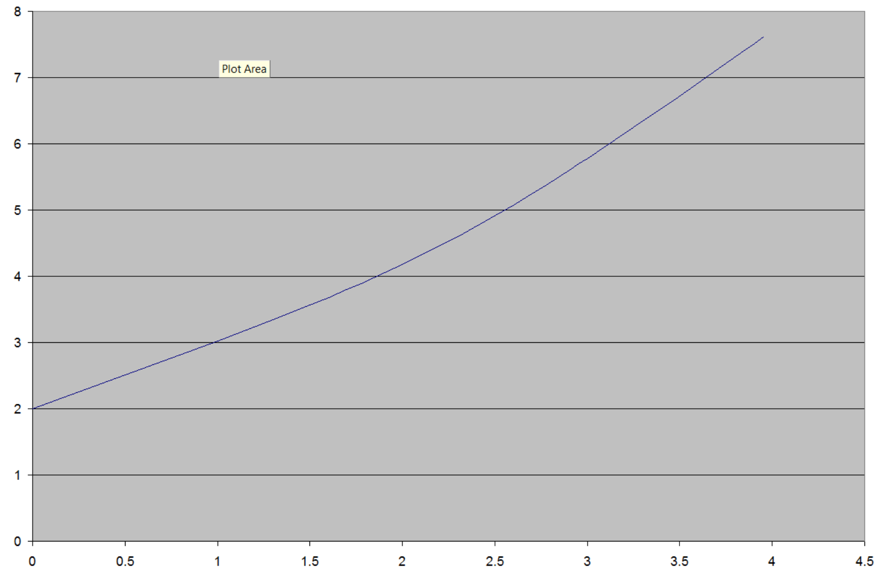
D2 f_x =LOG(B2)

	A	B	C	D
1	1	101	0	2.004321
2	10	1055	1	3.023252
3	20	2210	1.30103	3.344392
4	30	3465	1.477121	3.539703
5	40	4820	1.60206	3.683047
6	50	6275	1.69897	3.797614
7	60	7830	1.778151	3.893762
8	70	9485	1.845098	3.977037
9	80	11240	1.90309	4.050766
10	90	13095	1.954243	4.117106
11	100	15050	2	4.177536
12	200	40100	2.30103	4.603144
13	300	75150	2.477121	4.875929
14	400	120200	2.60206	5.079904
15	500	175250	2.69897	5.243658
16	600	240300	2.778151	5.380754
17	700	315350	2.845098	5.498793
18	800	400400	2.90309	5.602494
19	900	495450	2.954243	5.695
20	1000	600500	3	5.778513
21	2000	2201000	3.30103	6.34262
22	3000	4801500	3.477121	6.681377
23	4000	8402000	3.60206	6.924383
24	5000	13002500	3.69897	7.114027
25	6000	18603000	3.778151	7.269583
26	7000	25203500	3.845098	7.401461
27	8000	32804000	3.90309	7.515927
28	9000	41404500	3.954243	7.617048
29				

We can verify this visually with a plot of C vs D. Select C and D, rows 1 through 28:

	A	B	C	D
1	1	101	0	2.004321
2	10	1055	1	3.023252
3	20	2210	1.30103	3.344392
4	30	3465	1.477121	3.539703
5	40	4820	1.60206	3.683047
6	50	6275	1.69897	3.797614
7	60	7830	1.778151	3.893762
8	70	9485	1.845098	3.977037
9	80	11240	1.90309	4.050766
10	90	13095	1.954243	4.117106
11	100	15050	2	4.177536
12	200	40100	2.30103	4.603144
13	300	75150	2.477121	4.875929
14	400	120200	2.60206	5.079904
15	500	175250	2.69897	5.243658
16	600	240300	2.778151	5.380754
17	700	315350	2.845098	5.498793
18	800	400400	2.90309	5.602494
19	900	495450	2.954243	5.695
20	1000	600500	3	5.778513
21	2000	2201000	3.30103	6.34262
22	3000	4801500	3.477121	6.681377
23	4000	8402000	3.60206	6.924383
24	5000	13002500	3.69897	7.114027
25	6000	18603000	3.778151	7.269583
26	7000	25203500	3.845098	7.401461
27	8000	32804000	3.90309	7.515927
28	9000	41404500	3.954243	7.617048

Then choose Insert → Scatter graph and get this plot, which looks straight at first, then shifts to a different straight line on the right side. This right side is the high values of n, and that's the behavior we're interested in:



It occurs when column C is 3 or more. So, let's ask Excel to give us the parameters of a straight line for that part of the data. In cell F20 (could be in any cell, actually), I'll type
`=slope(D20:D28,C20:C28)` .
(The D's are the y coords, and the C's are the x coords). And, in cell next to it, `G20`, I'll type
`=intercept(D20:D28,C20:C28)`.

Excel is telling us that, if we were to fit a least-squares straight line to the data in rows 20 to 28 of columns C and D, we would get
$\log(C(n)) = m\log(n) + b$
with m = 1.93105176 and b = -0.025516668. But $b = \log(A)$, and if we compute `=10`$^{\texttt{G20}}$ we'll get A, which is 0.942938422.

Thus our empirical fit to the function $C(n)$ is
$C(n) \approx 0.9429 \times n^{1.931}$.

Does this work for n=9000? Pluggin in, we get 40769021, which is close to 41404500, the true value.