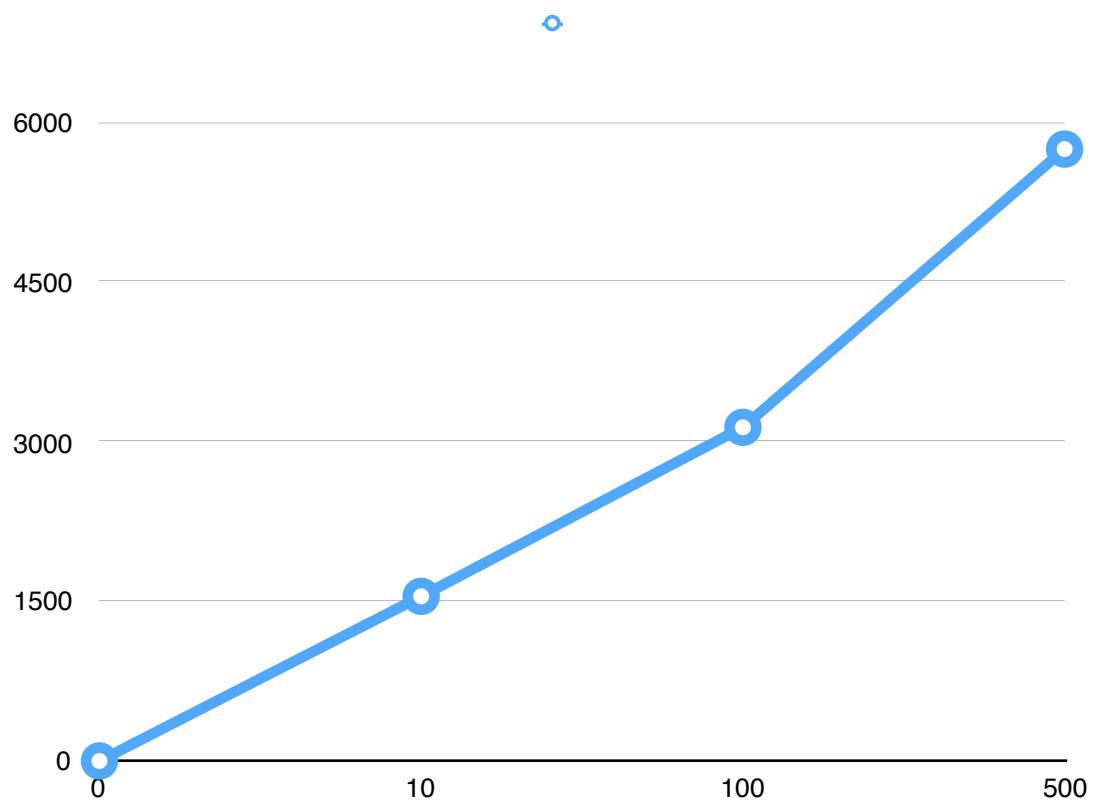


Ex1.

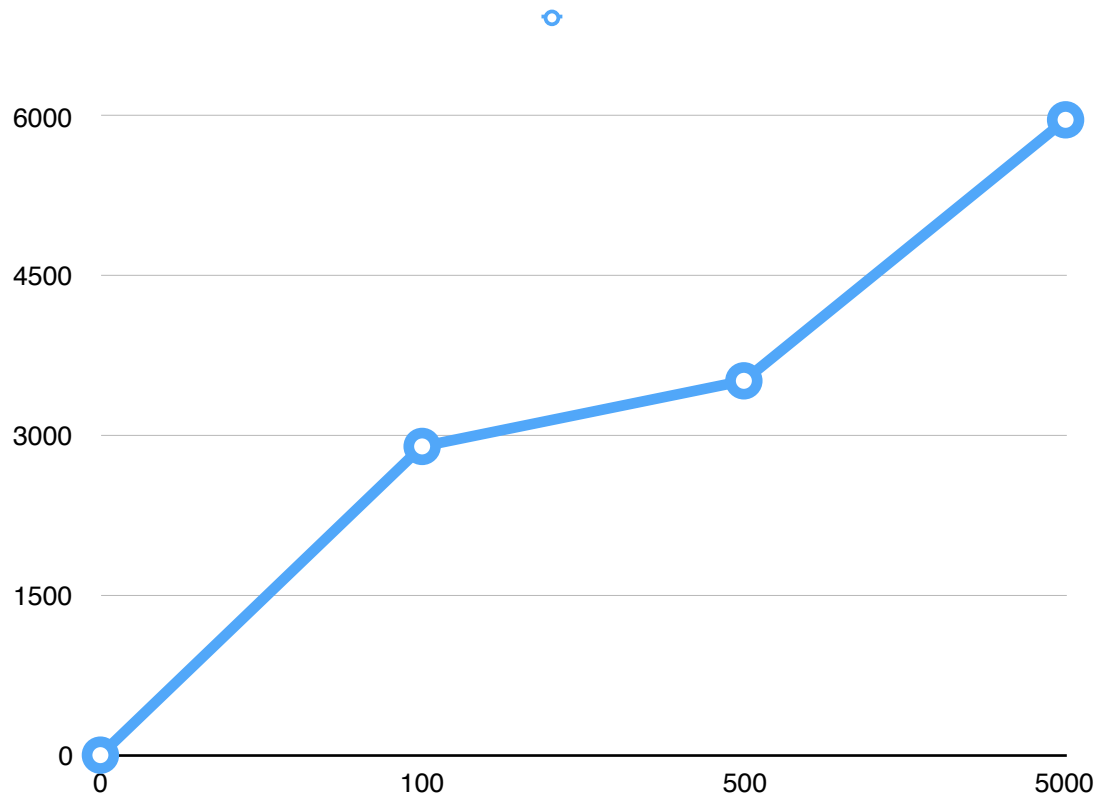
Degree of polynomial(n)	Evaluation Time(millisecs)
10	1546
100	3133
500	5746



Ex2

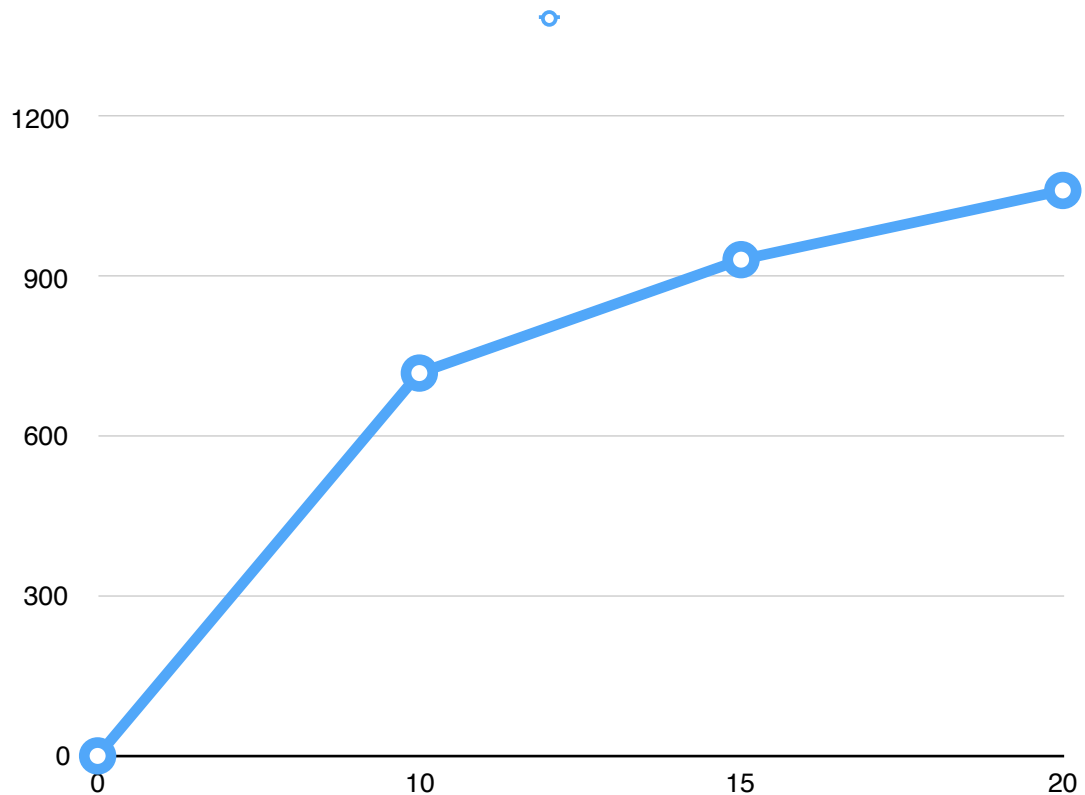
Size of matrix(n)	Execution time(millisecond)
100	2899

Size of matrix(n)	Execution time(millisec)
500	3515
5000	5966



Ex3

Value of n	Execution Time to generate 2 power n binary numbers
10	719
15	932
20	1062



Ex4 A

Code:

```

16 for(int i=a.length();i>0;i-)—————first loop O(n)
17 {
18     if(a.charAt(c-i)==b.charAt(0))
19     {
20         if(b.length()==1){result+=" "+(c-i);}
21         for(int n=b.length()-1;n>0;n-)————second loop O(n)
22         {
23             int count=1;
24             if(b.charAt(d-n)!=a.charAt(c-i+count))
25             {
26                 break;
27             }
28             count++;
29             result+=" "+(c-i);
30         }
31     }
32 }

```

So the complexity of the code is $O(n*n)=O(n^2)$

EX4 B

Substring	Number of matches found	Time taken(ms)
T	10792	1576
CC	1552	4822
GCT	0	4861
GG	1565	3575
TCCT	0	5271
CTCTGCTCGCGCT	0	7831
GCTCGCCGT	0	7069

In my case, the longer substring is, the more time it will take to evaluate the data. I tried for so many times but the output remained same like the table above. Maybe it is because of my algorithm is different from other people. In my code, program has to compare two strings one char by one char. So it might take more time to compare a longer string.