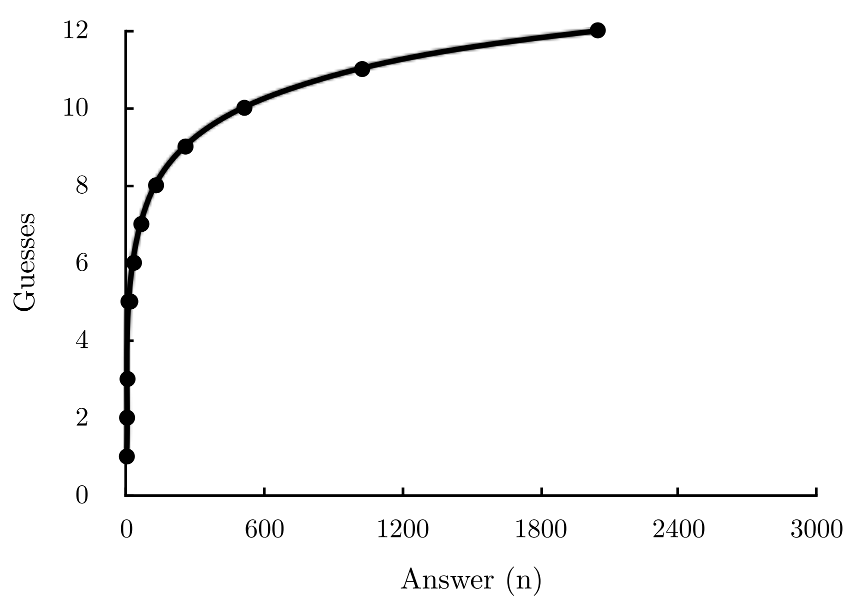
1.c.i

Table 2^k

RANGE	ANSWER (n)	GUESSES	RESULT OF FORMULA
[1:1]	1	1	$2^k = 1$
[1:2]	2	2	$2^k = 2$
[1:4]	4	3	$2^k = 3$
[1:8]	8	5	$2^k = 4$
[1:16]	16	5	$2^k = 5$
[1:32]	32	6	$2^k = 6$
[1:64]	64	7	$2^k = 7$
[1:128]	128	8	$2^k = 8$
[1:256]	256	9	$2^k = 9$
[1:512]	512	10	$2^k = 10$
[1:1024]	1024	11	$2^k = 11$
[1:2048]	2048	12	$2^k = 12$

$$log_2(n) + 1$$

Graph 2^k



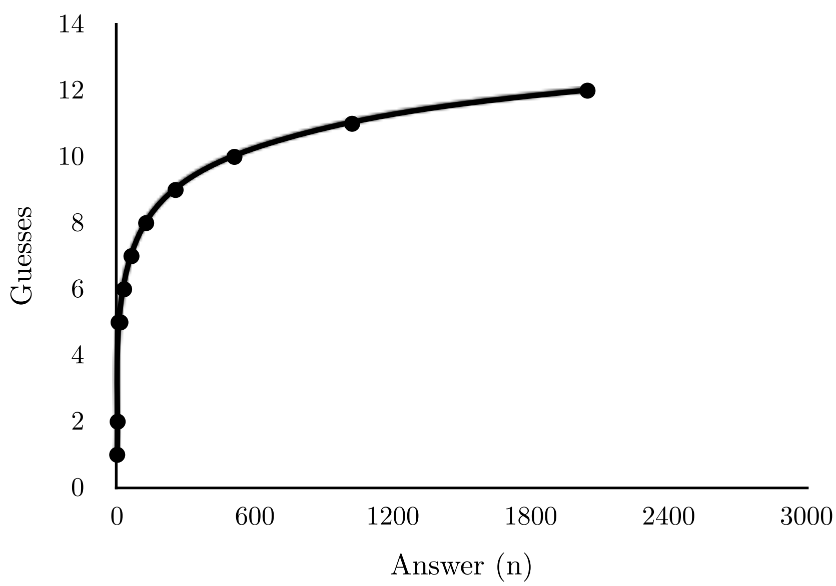
1.c.ii

Table 2^k - 1

RANGE	ANSWER (n)	GUESSES	RESULT OF FORMULA
[1:1]	1	1	$2^k - 1 = 1$
[1:3]	3	2	$2^k - 1 = 2$
[1:7]	7	5	$2^k - 1 = 4$
[1:15]	15	5	$2^k - 1 = 5$
[1:31]	31	6	$2^k - 1 = 6$
[1:63]	63	7	$2^k - 1 = 7$
[1:127]	127	8	$2^k - 1 = 8$
[1:255]	255	9	$2^k - 1 = 9$
[1:511]	511	10	$2^k - 1 = 10$
[1:1023]	1023	11	$2^k - 1 = 11$
[1:2047]	2047	12	$2^k - 1 = 12$

$$\lceil log_2(n)+1 \rceil$$

Graph 2^k - 1



1.d

$O(\log_2 n)$

2

$f(n) = \lceil n/2 \rceil - 1$
$O(n)$

3

```
remove(vector v, int k){
      v[k] = v[v.size-1];
      remove v[v.size-1];
      v.size = v.size - 1;
}
```

4

Mathematically, n^2 will be less than *nlog(n) as long as n is less than 100.* This means that for cases when n is less than 100, O(n^2) is faster than O(*nlog(n)*). n^2 will be larger than *nlog(n) as long as n is greater than 100.* This means that for cases when n is greater than 100, O(n^2) is slower than O(*nlog(n)*).

5

1: $f(n) = n - 1$
$O(n)$

2: $f(n) = 4n^2 + 1$
$O(n^2)$

3: $f(n) = n^2 - n + 3$
$O(n^2)$