CSCE 221-200: Honors Data Structures and Algorithms
Assignment Cover Page
Spring 2021

Name:	Priyanshu Barnwal
Email:	Priyanshu@tamu.edu
Assignment:	Homework 1
Grade (filled in by grader):	

Please list below all sources (people, books, webpages, etc) consulted regarding this assignment (use the back if necessary):

CSCE 221 Students	**Other People**	**Printed Material**	**Web Material (give URL)**	**Other Sources**
1.	1.	1. The Textbook	1. https://stackoverflow.com/questions/18466309/recursion-counter-inside-a-c-function	1.
2.	2.	2.	2. https://www.programiz.com/cpp-programming/recursion	2.
3.	3.	3.	3. http://www.cplusplus.com/reference/string/to_string/	3.
4.	4.	4.	4. https://www.instructables.com/How-to-Convert-Numbers-to-Binary/	4.
5.	5.	5.	5. https://stackoverflow.com/questions/33194931/when-can-an-algorithm-have-square-rootn-time-complexity#:~:text=6%20Answers&text=Square%20root%20time%20complexity%20means,which%20takes%20that%20much%20time.	5.
			https://www.google.com/search?q=min+to+ms&rlz=1C1RXQR_enUS931US931&oq=min+to+ms&aqs=chrome..69i57j6j0l4.5890j1j7&sourceid=chrome&ie=UTF-8	

Recall that TAMU Student Rules define academic misconduct to include acquiring

answers from any unauthorized source, working with another person when not specifically permitted, observing the work of other students during any exam, providing answers when not specifically authorized to do so, informing any person of the contents of an exam prior to the exam, and failing to credit sources used. *Disciplinary actions range from grade penalty to expulsion.*

"On my honor, as an Aggie, I have neither given nor received unauthorized aid on this academic work. In particular, I certify that I have listed above all the sources that I consulted regarding this assignment, and that I have not received or given any assistance that is contrary to the letter or the spirit of the collaboration guidelines for this assignment."

Signature:	Priyanshu Barnwal
Date:	1/24/2021

```
#include <iostream>
#include <string>
#include <cmath>
using namespace std;

int power_two(int num) {
	int last = 0;
	int rem = num % 2;

	last += rem;
	if (num != 1)
		last += (power_two(num / 2)*10);

	return last;
}
```

Exercise 2.7a:

1. O(n)
2. O(n^2)
3. O(n^3)
4. O(n^2)
5. O(n^3)
6. O(n^4)

Exercise 2.12:

1. 12 million
2. 3,656,807.36
3. 34641.02
4. 4932.42

Exercise 2.20:

1.
```
bool prime(int x) {
	for (int n = 2; n < x / 2; n++)
		if (x % n == 0)
			return false;

		return true;

}
```
2. O(N) is the worst case run time because there is a for loop. I am not sure how to lower this.
3. $O(\log_2(N))$
4. $2^{B/2}$
5. T^2 and T^3
6. In terms of B because it is more accurate.

Exercise 2.23:

```
long long power(long long x, int n) {
	int number = x;
	if (n == 0) {
		return 1;
	}
	if (n == 1) {
		return x;
	}

	if (n % 2 == 0)
		for (int i = 0; i < n / 2; i++)
			x *= x;
	else {
		for (int i = 0; i < n / 2; i++) {
			cout << n << endl;
			x *= x;
		}
		x *= number;
	}
	return x;
}
```