CSCE 221 Cover Page
Homework #1
Due July 12 at midnight to eCampus

First Name Laura Last Name Austin UIN 524006473

User Name laustin254 E-mail address laustin254@tamu.edu

Please list all sources in the table below including web pages which you used to solve or implement the current homework. If you fail to cite sources you can get a lower number of points or even zero, read more: Aggie Honor System Office

Type of sources		
People	I worked with Aaron Ingram I worked with Callen McCauley	
Web pages (provide URL)	https://stackoverflow.com/questions/8480640/how-to-throw-a-c-exception	http://www.cplu
Printed material	"Data Structures and Algorithms" textbook for the class	
Other Sources		

I certify that I have listed all the sources that I used to develop the solutions/codes to the submitted work.

"On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work."

Your Name Laura Austin Date July 12, 2017

1.

part (a) Ensure that if the vector is not sorted, then exception is thrown.

```
[laustin254]@linux2 ~/HW1> (19:42:49 07/12/17)
:: ./a.out BinarySearch
vector tested is: < 2 1 4 8 16 32 64 128 256 512 1024 2048 >
terminate called after throwing an instance of 'std::invalid_argument'
  what():  Input vector not sorted
Aborted
```

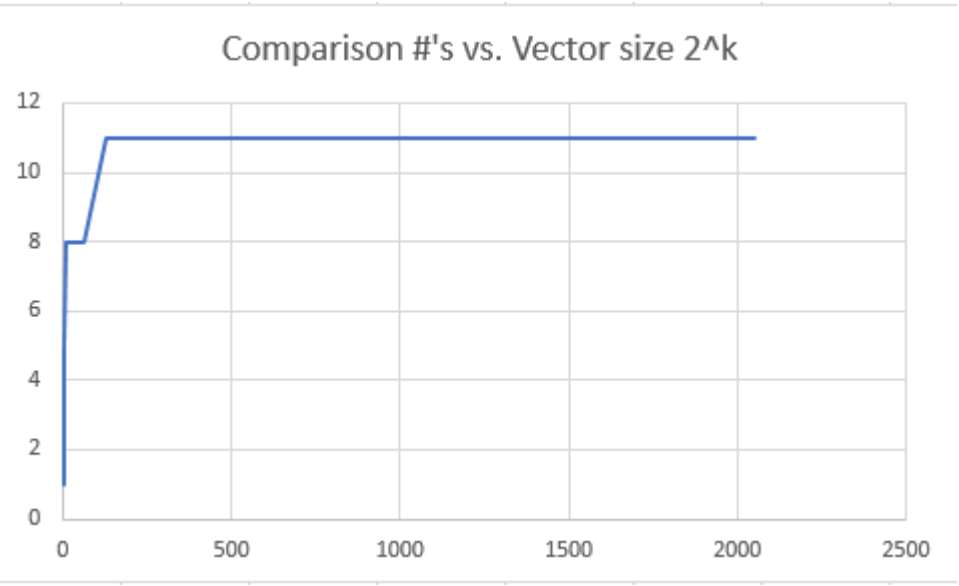
part (b) Searching both the ascending and descending vectors.

```
[laustin254]@linux2 ~/HW1> (19:38:55 07/12/17)
:: ./a.out BinarySearch
vector tested is: < 2048 1024 512 256 128 64 32 16 8 4 2 1 >
found element 1 after 11 comparisons.

[laustin254]@linux2 ~/HW1> (19:38:55 07/12/17)
:: g++ -std=c++11 BinarySearch.cpp

[laustin254]@linux2 ~/HW1> (19:39:19 07/12/17)
:: ./a.out BinarySearch
vector tested is: < 1 2 4 8 16 32 64 128 256 512 1024 2048 >
 found element 2048 after 11 comparisons.
```

part (c)

Range [1,n]	Target for incr. values	# comp. for incr. values	Target for decr. values	# comp. for decr. values	Result of the formula in item 5 (e)
[1,1]	1	2	1	2	
[1,2]	2	5	1	5	
[1,4]	4	5	1	5	
[1,8]	8	8	1	8	
[1,16]	16	8	1	8	
[1,32]	32	8	1	8	
[1,64]	64	8	1	8	
[1,128]	128	11	1	11	
[1,256]	256	11	1	11	
[1,512]	512	11	1	11	
[1,1024]	1024	11	1	11	
[1,2048]	2048	11	1	11	

part (d)



part (e) $f(n) = 2 + log_2 n$

part (f) Changing vector to 2^k - 1.

Range [1,n]	Target for incr. values	# comp. for incr. values	Target for decr. values	# comp. for decr. values	Result of the formula in item 5 (e)
[1,1]	1	2	1	2	
[1,3]	3	5	1	5	
[1,7]	7	5	1	5	
[1,15]	15	8	1	8	
[1,31]	31	8	1	8	
[1,63]	63	8	1	8	
[1,127]	127	8	1	8	
[1,255]	255	11	1	11	
[1,511]	511	11	1	11	
[1,1023]	1023	11	1	11	
[1,2047]	2047	11	1	11	

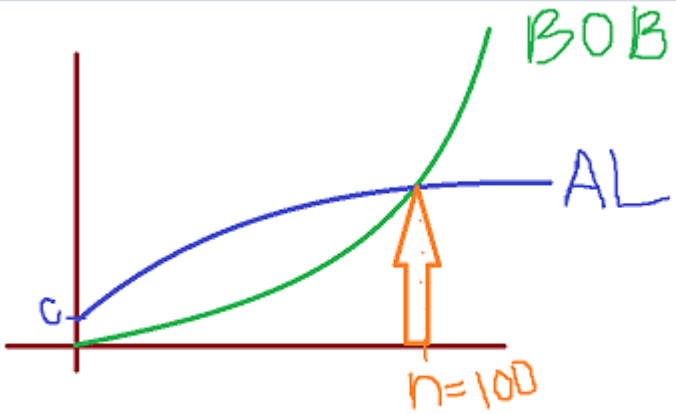
part (g) $O(logn)$

2. (problem 4.7) $8nlogn$ will be faster than $2n^2$, however up until n_0, $2n^2$ will be faster. To find n_0, setting both equations equal to each other will determine when they equal, and at what value of n,that this occurs. $8nlogn = 2n^2 \implies 4logn = n$. Assuming that this is a logarithm base 2, then $n/log_2 n = 4 \implies 16/log_2 16 = 16/4 = 4$. So $n = 16$. For every value after $n = 16$, algorithm A will be better, so for all $n > 16$.

3. (problem 4.21) The worst case run time in terms of n is: $O(n^2)$

In terms of in, then it'd be $O(n^2)$. This is not a linear algorithm because $n^2$is quadratic.

4. (problem 4.39) The cutoff mark $n = 100$, basically means that the following graph applies:



since $nlogn$ does not crossn^2, we know that Al's $f(n)$ formula must be ADDED to some constant, in other words, Al's $f(n) = nlogn + C$, whereas Bob's $f(n) = n^2$. Also that is an image I drew in paint. copyright.

5.$s = n$.
$f(n) = 2(n-1) + 1 \Longrightarrow O(n)$
$f(n) = 2(n-2) + 1 \Longrightarrow O(n)$
$f(n) = 2(n-1)(2)(n) + 1 = 4n(n-1) + 1 \Longrightarrow O(n^2)$
$f(n) = (2+2)(n-1) + 2 = 4n - 4 + 2 = 4n - 2 \Longrightarrow O(n)$