

Name: (as it would appear on official course roster)	
Umail address:	@umail.ucsb.edu
Optional: name you wish to be called if different from name above.	
Optional: name of "homework buddy" (leaving this blank signifies "I worked alone")	

1

h00

CS32 W19

h00: Linear Search

ready?	assigned	due	points
true	Mon 01/07 09:30AM	Mon 01/14 09:30AM	50

You may collaborate on this homework with AT MOST one person, an optional "homework buddy".

MAY ONLY BE TURNED IN IN THE LECTURE/LAB LISTED ABOVE AS THE DUE DATE, OR IF APPLICABLE, SUBMITTED ON GRADESCOPE. There is NO MAKEUP for missed assignments; in place of that, we drop the lowest scores (if you have zeros, those are the lowest scores.)

1. (10 pts) Before attending your first discussion section (lab), please ensure that you can access your CSIL / College of Engineering Account. Either visit CSIL (the lab on first floor ocean side of Harold Frank Hall), and try to login, or use an ssh client to see if you can access csil.cs.ucsb.edu.

If you need help creating or accessing your CoE/CSIL account, visit <https://accounts.engr.ucsb.edu>

Then, write your College of Engineering account username below. DO NOT WRITE YOUR PASSWORD. DO NOT EVER WRITE YOUR PASSWORD!!

Course Textbooks

- **PS9: Problem Solving with C++, 9th Edition** by Walter Savitch (used in CS16 - Purple/Blue cover. There is a newer edition available, but I don't expect everyone to buy a new edition if you already have the 9th edition from when you took CS 16).
- **DS4: Data Structures and Other Objects Using C++, 4th edition** by Michael Main and Walter Savitch (used in CS24 - Brown cover)

Both books are required for this course. This homework will only refer to DS4 (the brown book), but future homework assignments may draw on either or both of the two books.

Reading: In DS4 read pages 584-594.

Also, read the "C++ Feature" box about `size_t` on p. 100. You only need to read the part in the sans-serif font beside the "C++ Feature" logo, but if you read a few extra lines of the main text, you'll see better how `size_t` is used. Since `size_t` is used in Section 12.1, it's helpful to review this material from Chapter 3 in case you glossed over it during CS24.

Then, answer the questions on the back of this homework.

Questions about DS4, pages 584-586.

2. (10 pts) Serial search is also called "linearSearch". Suppose you are writing a standalone C++ function called **linearSearch** that searches through an unsorted C++ builtin array of int values using this technique. Write the function prototype for this function—ONLY the function prototype. Be sure to include the name and type all the input parameters you would need, and a suitable type for the return value. Assume the function returns the index of the value if found, and -1 if the value is not found. Choose parameter names that make sense and represent good programming style (i.e. clearly document the purpose of the parameters that are being passed in.)

2

h00

CS32 W19

3. (10 pts) For this question, please review pp 584-586 and read both sides of the handout (<http://www.cs.ucsb.edu/~richert/cs32/misc/h00-handout.pdf>). Then answer this question about the expected number of array accesses for linear search.

- Suppose there are nine array elements, $a[0]$ through $a[8]$.
- Each of the odd elements, $a[1]$, $a[3]$, $a[5]$, and $a[7]$ has a 10% chance (0.1) of being the one searched for.
- Each of the even elements $a[0]$, $a[2]$, $a[4]$, $a[6]$ and $a[8]$ has a 5% change (0.05) of being the one searched for.
- With probability 35% (0.35), the element being sought is not found in the array.

In this case, what is the expected number of array accesses?

4. In CMPSC 24, you should have learned about big-Oh analysis, i.e. asymptotic worst-case run-time analysis.

- a. (10 pts) What is the big-O analysis of a `find(key)` operation on a sorted array of n keys, implemented using *binary search*? Circle your answer. Note that \log , in the context of big-O, typically means log based 2, but since all logs are related by a constant factor, i.e. for any value of b , there is a constant C such that $\log_2(x) = C \cdot \log_b(x)$.

$O(1)$

$O(\log n)$

$O(n)$

$O(n \log n)$

$O(n^2)$

$O(n^2 \log n)$

$O(n^3)$

- b. (10 pts) Explain why this is the running time for binary search. A "formal proof" is not required—just a *brief* intuitive explanation that shows you understand why this is the running time.