| Name: | | |
|---|---|---|
| *(as it would appear on official course roster)* | | |
| Umail address: | @umail.ucsb.edu | **section** 9am, 10am, 11am, 12pm |
| 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**

**h06**

**CS32 F19**

# h06: Binary Search Review

| ready? | assigned | due | points |
|---|---|---|---|
| true | Tue 10/15 02:00PM | Tue 10/22 02:00PM | 50 |

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

MAY ONLY BE TURNED IN IN THE LECTURE 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.)

---

**Reading: Binary Search Review, DS 12.1, Review 2.6, 6.1**

1. (10 pts) Fill in the information in the header. The following are required to get the 10 "participation" points.
   - Filling in your name and umail address.

   Also: For paper submission PLEASE submit on ONE SHEET OF PAPER, double-sided if at all possible. If you must submit on two printed sheets write name on BOTH sheets and no staples, paperclips, or folded corners.

   Please also read the handout http://cs.ucsb.edu/~richert/cs32/misc/h06-handout.pdf

   Note typo: The image at right shows a CORRECTED version of the formula for the halving function from p. 593. (The log term should have the floor function applied to it—in some printings of the textbook, this is missing due to a typo.)

   **Value of the Halving Function**

   $H(n)$ = (the number of times that $n$ can be divided by 2, stopping when the result is less than 1) has the value:
   $$H(n) = \lfloor \log_2 n \rfloor + 1$$

2. (10 pts) Given the definition of H(n) above, give the value of each of these expressions:

   H(6) =　　　　　　　　　H(7) =　　　　　　　　　H(8) =

   H(1,000) =　　　　　　　H(1,000,000) =

3. (10 pts) Read over pg. 86 about the STL pair template class, as well as the information on page 1 of the handout http://cs.ucsb.edu/~richert/cs32/misc/h06-handout.pdf . Then write the function definition for `distanceBetween` in the space below. You may assume that the `#include` directives for math andutility, as well as the `using namespace std;` statement has already appeared in the file.

**2**

**h06**

**CS32 F19**

4. (10 pts) Below, please show the steps of the binary search algorithm when searching for 55 in the array shown. Show the steps involved. Note that 55 **is** in the array. Note that the algorithm as shown in the book must do an actual recursive call on an array of size 0 before it determines that the element is not found. Show that call as well. An example of a solved problem is given in the handout: http://cs.ucsb.edu/~richert/cs32/misc/h06-handout.pdf:

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| target=55 | | | | | | | | | | | | | | | | | | | |
| step | values passed in | | middle (first + size/2) | array values | | | | | | | | | | | | values passed to next step | |
| | first passed in | size passed in | | a[0] | a[1] | a[2] | a[3] | a[4] | a[5] | a[6] | a[7] | a[8] | a[9] | a[10] | | | first = first, or middle+1 | size = size/2, or (size-1)/2 |
| step 1 | 0 | 11 | 5 = 0 + (11/2) | 13 | 21 | 34 | 41 | 55 | 66 | 72 | 86 | 94 | 107 | 118 | | |
| step 2 | | | | | | | | | | | | | | | | |
| step 3 | | | | | | | | | | | | | | | | |
| step 4 | | | | | | | | | | | | | | | | |
| step 5 | | | | | | | | | | | | | | | | |

5. (10 pts) Below, please show the steps of the binary search algorithm when searching for 15 in the array shown. Show the steps involved. Note that 15 **is not** in the array. Note that the algorithm as shown in the book must do an actual recursive call on an array of size 0 before it determines that the element is not found. Show that call as well. An example of a solved problem is given in the handout: http://cs.ucsb.edu/~richert/cs32/misc/h06-handout.pdf:

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| target=15 | | | | | | | | | | | | | | | | | | | |
| step | values passed in | | middle (first + size/2) | array values | | | | | | | | | | | | values passed to next step | |
| | first passed in | size passed in | | a[0] | a[1] | a[2] | a[3] | a[4] | a[5] | a[6] | a[7] | a[8] | a[9] | a[10] | | | first = first, or middle+1 | size = size/2, or (size-1)/2 |
| step 1 | 0 | 11 | 5 = 0 + (11/2) | 13 | 21 | 34 | 41 | 55 | 66 | 72 | 86 | 94 | 107 | 118 | | |
| step 2 | | | | | | | | | | | | | | | | |
| step 3 | | | | | | | | | | | | | | | | |
| step 4 | | | | | | | | | | | | | | | | |
| step 5 | | | | | | | | | | | | | | | | |