

---

## Practice Exam 1

- Do not open this exam booklet until you are directed to do so. Read all the instructions on this page.
- When the exam begins, write your name on every page of this exam booklet.
- This exam contains 5 problems, some with multiple parts. You have 120 minutes to earn 90 points.
- This exam booklet contains 10 pages, including this one. Two extra sheets of scratch paper are attached. Please detach them before turning in your exam at the end of the examination period.
- This exam is closed book. You may use one handwritten  $8\frac{1}{2} \times 11$  or A4 crib sheet. No calculators or programmable devices are permitted.
- Write your solutions in the space provided. If you need more space, write on the back of the sheet containing the problem. Do not put part of the answer to one problem on the back of the sheet for another problem, since the pages may be separated for grading.
- Do not waste time and paper rederiving facts that we have studied. It is sufficient to cite known results.
- Do not spend too much time on any one problem. Read them all through first, and attack them in the order that allows you to make the most progress.
- Show your work, as partial credit will be given. You will be graded not only on the correctness of your answer, but also on the clarity with which you express it. Be neat.
- Good luck!

Name: \_\_\_\_\_ ID: \_\_\_\_\_

Section you want your test returned in (circle one):      001      002

Problem	1	2	3	4	5	Total
Points	12	18	20	15	25	90
Grade						

Name: \_\_\_\_\_

CSE 465 Practice Exam 1, page 2

**Problem 1** (Big- $O$  Notation, **12 points**). Rank the following functions by increasing order of growth, that is, find an arrangement  $g_1, \dots, g_{12}$  of the functions satisfying  $g_1(n) = O(g_2(n))$ ,  $g_2(n) = O(g_3(n))$ , .... Break the functions into classes so that  $f$  and  $g$  are in the same class if and only if  $f(n) = \Theta(g(n))$ . Note that  $\log(\cdot)$  is the base 2 logarithm and  $\log_b(\cdot)$  is the base  $b$  logarithm.

$$\sum_{i=1}^{3n} (2i + 1), \log_3(n^2), 2^n, n^{\frac{1}{465}}, n^{465},$$
$$n \log n, 3^{\log n}, n^{\log n}, \log(n!), n!, n^n, n^{\log_2 3}$$

**Problem 2** (Recurrences, **18 points**). Solve the recurrences in parts **(a)** to **(d)**, expressing your answers using  $\Theta$ -notation. Whenever possible, apply the Master Theorem and state which case you used. If the Master Theorem does not apply, (i) draw a recursion tree, (ii) specify its height, (iii) estimate the sum of the nodes at each level, and (iv) give the solution to the recurrence.

Assume that  $T(n) = \Theta(1)$  for small  $n$ , and that (when applicable) the Regularity Condition is met.

**(a)**  $T(n) = 3T(n/3) + \sqrt{n}$

**(b)**  $T(n) = 4T(n/2) + 3n^2$

**(c)**  $T(n) = T(n^{1/2}) + \log \log n$

(Continued on next page.)

Name: \_\_\_\_\_ CSE 465 Practice Exam 1, page 4

(d)  $T(n) = 4T(n/5) + n^2$

Use the substitution method to prove  $T(n) = \Omega(n^2)$  for  $T(n)$  satisfying:

(e)  $T(n) = T(n/2) + T(4n/5) + T(7n/10) + n$

**Problem 3** (True or False, and Justify, **20 points**). Circle **T** or **F** for each of the following statements to indicate whether the statement is true or false, respectively. If the statement is correct, briefly state why. If the statement is wrong, explain why. The better your argument, the higher your grade, but be brief. **No points will be given even for a correct solution if no justification is presented.**

**T F** For all asymptotically nonnegative functions  $f$ ,  $f(n) + o(f(n)) = \Theta(f(n))$ .

**T F** Insertion Sort takes  $O(n)$  time on the following input of length  $n$ :

$$\lfloor \frac{n}{2} \rfloor + 1, \lfloor \frac{n}{2} \rfloor + 2, \dots, n - 1, n, 1, 2, \dots, \lfloor \frac{n}{2} \rfloor - 1, \lfloor \frac{n}{2} \rfloor.$$

**T F** The recursion tree for Mergesort, on an array of size  $n$ , has  $n \log n$  leaves.

**T F** Recall that `RANDPARTITION` is a subroutine used in the Quick Sort and Randomized Selection. Recall that we say that the split produced by `RANDPARTITION` is *OK* if the array of size  $n$  is split into two parts of size at least  $n/4$  each. Let  $I$  be the indicator variable for the event that `RANDPARTITION` gives an *OK* split. Then the expectation of  $I$  is 2.

**Problem 4** (Loop invariants, **15 points**). Recall the following problem from Homework 3:

You are consulting for a small investment company. They give you a price of Google's shares for the last  $n$  days. Let  $p(i)$  represent the price for day  $i$ . During this time period, the company wanted to buy 1,000 shares on some day and sell all these shares on some later day. The company wants to know when they should have bought and when they should have sold the shares in order to maximize the profit. If there was no way to make money during the  $n$  days, you should report this instead.

For example, suppose  $n = 3, p(1) = 9, p(2) = 1, p(3) = 5$ . Then you should return "buy on day 2, sell on day 3".

You mention the problem to Professor Onepass, and she suggests the following algorithm:

BESTTWO DAYS( $p$ )

```

    ▷ Prices are given in the array  $p$ 
1   $n \leftarrow \text{length}(p)$ 
2  if  $n = 1$ 
3      then return "No way to make money."
4   $buy \leftarrow 1$ 
5   $sell \leftarrow 2$ 
6   $minsofar \leftarrow \min(p[1], p[2])$ 
7  for  $k \leftarrow 3$  to  $n$ 
8      do if  $(p[k] - p[minsofar]) > (p[sell] - p[buy])$ 
9          then  $sell \leftarrow k$ 
10          $buy \leftarrow minsofar$ 
11         if  $p[k] < p[minsofar]$ 
12             then  $minsofar \leftarrow k$ 
13 if  $p[buy] \geq p[sell]$ 
14     then return "No way to make money."
15     else return "Buy on day "  $buy$  ", sell on day "  $sell$ 
```

(a) Give the running time of the code using asymptotic notation.

(b) State a loop invariant for the **for** loop in lines 7–12.

Name: \_\_\_\_\_ CSE 465 Practice Exam 1, page 7

(c) Prove that the algorithm is correct using your loop invariant.

(i) Initialization

(ii) Maintenance

(iii) Termination

**Problem 5** (Divide and Conquer, **25 points**). A *triomino* is an *L*-shaped tile formed by 1-by-1 adjacent squares. The problem is to cover any  $2^b$ -by- $2^b$  chessboard with one missing square (anywhere on the board) with triominos. Triominos should cover all squares except the missing one with no overlaps.

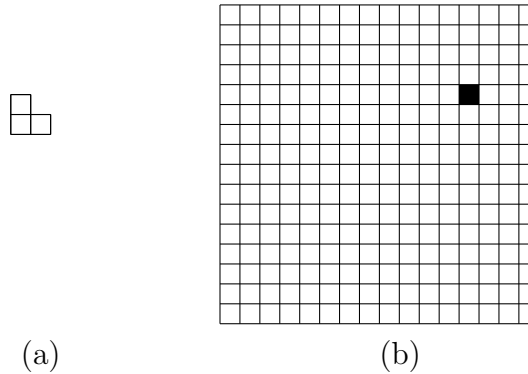


Figure 1: (a) A triomino tile. (b) A  $16 \times 16$  chessboard with one missing square.

- (a) Design a divide-and-conquer algorithm for this problem. The inputs are  $b$  (which determines the size of the board) and the coordinates  $(x, y) \in \{1, \dots, 2^b\} \times \{1, \dots, 2^b\}$  of the missing square. The output should be a list of triples, where each triple describes the position of one of the triominos. First, explain your algorithm concisely in English (feel free to use pictures). Second, specify your algorithm using pseudocode.

(Continued on next page)

Name: \_\_\_\_\_ CSE 465 Practice Exam 1, page 9

Problem 5 (a) continued.

(Continued on next page)

Name: \_\_\_\_\_ CSE 465 Practice Exam 1, page 10

(b) Explain in a couple of sentences why your algorithm is correct.

(c) Give a recurrence for the worst case running time of your algorithm in terms of  $b$  and solve it. How long does your algorithm take as a function of  $b$ ?