

CS 324 Homework Assignment 1

Due: 11:59pm, Thursday, September 21st

This assignment is scored out of 53. It consists of 7 questions. When you submit, you are required to create a folder with your name (Last name first, then First name), CS324, HW1, e.g., LastName_FirstName_CS324_HW1. Type your answers into a text file (**only .txt, .doc, and .pdf file formats are accepted**) and save it in this folder. Put all your Java programs (***.java**) as well as output files in the same folder. Zip this folder, and submit it as one file to Desire2Learn. Do not hand in any printouts. Triple check your assignment before you submit. **If you submit multiple times, only your latest version will be graded and its timestamp will be used to determine whether a late penalty should be applied.**

Short Answers

P1. (3pts) Under what circumstances, when a searching operation is needed, would Sequential Search (Algorithm 1.1 in the textbook) not be appropriate?

P2. (6pts, 2pts each) Consider the following algorithm:

```
void myfun1(int n) {
    System.out.println("n = " + n);
    for (int k = 1; k <= n / 2; k++) {
        System.out.println(k);
        for (int m = 1; m <= k; m++)
            System.out.println(k + ", " + m);
    }
}
```

- Trace through and show the output when $n = 8$.
- What is the exact formula for $T(n) = \#$ of lines of output of “**myfun1**” on argument n ?
- Which of the following sets does the correct formula for $T(n)$ belong to?
(A) $\theta(1)$ (B) $\theta(\log n)$ (C) $\theta((\log n)^2)$ (D) $\theta(n)$ (E) $\theta(n \log n)$ (F) $\theta(n(\log n)^2)$
(G) $\theta(n^2)$ (H) $\theta(n^2 \log n)$ (I) $\theta(n^2(\log n)^2)$ (J) $\theta(n^3)$ (K) $\theta(n^3 \log n)$ (L) None of the above.

P3. (6pts, 2pts each) Consider the following algorithm:

```
int myfun2(int n) {
    int p, k, t;
    p = n;
    k = 10;
    while (p >= 1) {
        k = k + p;
        for (t = n; t >= 0; t = t - 3) {
            System.out.println(p + ", " + t);
            k++;
        }
        p = p / 3;
    }
    return k;
}
```

- Trace through and show the return value and the output when $n = 9$.
- Exactly how many lines of output will there be if **myfun2** is called with $n = 729$?
- Using the choices in the problem above, which is the running time $T(n)$ of “**myfun2**”?

P4. (5pts, 1pt each) Evaluate the following expressions:

- $\log_2(64)$.
- $\log_2(1024 \times 256)$.
- $\log_2(2^n)$.
- $\log_2(8^n)$.
- $\log_2(x)$, where $x = \log_2(256^{128})$

P5. (4pts, 2pts each) Consider the following algorithm:

```
int myfun3(int[] A, int n) {
    int i, j, s = 0;

    for (i = 0; i < n; i++)
        s = s + A[i];

    for (i = 0; i < n; i++)
        for (j = i + 1; j < n; j++)
            if (A[i] == A[j])
                return -s;

    return s;
}
```

- Using the θ (theta) notation, what is the **best-case** running time of **myfun3**?
- Using the θ (theta) notation, what is the **worst-case** running time of **myfun3**?

P6. (2pts) Using the θ (theta) notation, what is the running time of the function **myfun4** (in terms of n)?

```
boolean myfun4(int[] A, int n) {
    int i, j;
    for (i = 0; i < n; i++)
        for (j = i + 1; j < n; j++)
            if (A[i] == A[j])
                return false;
            else
                return true;
    return true;
}
```

Programming Questions

P7. (27pts)

a. Completing the Homework1 class

In this problem, you are required to implement the following methods listed in the **Homework1** class:

boolean unique(int[][] arr) – This method takes an integer array (you can assume that the row and column dimensions are equal) and returns **true** if all values are unique (i.e., there are no repeats) and **false** otherwise. This method should NOT change the content of the array.

boolean isSorted(int[] arr, int low, int high) – This method takes an integer array as well as two integers. It returns **true** if the array is sorted in nondecreasing order between the indices **low** and **high**, or **false** otherwise. You are required to implement this method using a **recursive** solution! **You should NOT change the content of the array parameter.**

int indexOfMin(int[] A, int low, int high) – This method takes as parameters an **int** array **A**, as well as two indices **low** and **high**. It uses a **divide-and-conquer** algorithm to find the smallest element in **A** located between **low** and **high**. **You should NOT change the content of the array parameter.**

Note that you are only supposed to touch the above three methods. You are NOT allowed to create any other methods, instance variables, or make any changes to methods other than the above three methods or files other than "Homework1.java". Points will be taken off if you fail to follow this rule.

b. Code Testing

You are provided with a test driver implemented by "**TestHomework1.java**" (**Do not make any changes to this file!**) so there is no need to write your own.

Since the code is incomplete, the program will not compile until you get the three methods implemented. Once you have completed the methods, you can run the test. You should create a plain text file named "**output.txt**", copy and paste the output (if your code crashes or does not compile, copy and paste the error messages) to this file and save it.

Grading Rubrics:

Code does not compile: -10

Code compiles but crashes when executed: -5

Changes were made to things other than the required method: -5

unique changes the content of the array parameter: -3

isSorted was implemented in a non-recursive way: -5

isSorted changes the contents of the array parameter: -3

indexOfMin was implemented in a non-recursive way: -5

indexOfMin was implemented in a recursive but not divide-and-conquer way: -5

indexOfMin changes the content of the array parameter: -3

Has output file: 5

Code passes 22 test cases: 22 (each test case worth 1 point)

Sample Output:

Test 1: unique - [Passed]

1 2 3

4 5 6

7 8 9

Expected: true

Yours: true

Test 2: unique - [Passed]

1 2 3

4 5 1

7 8 9

Expected: false

Yours: false

...

Test 20: indexOfMin(low = 7, high = 10) - [Passed]

Expected: 7

Yours: 7

Test 21: indexOfMin(low = 5, high = 6) - [Passed]

Expected: 5

Yours: 5

Test 22: indexOfMin(low = 10, high = 10) - [Passed]

Expected: 10

Yours: 10

Total test cases: 22

Correct: 22

Wrong: 0