Final Exam

Warning: The hard deadline has passed. You can attempt it, but **you will not get credit for it**. You are welcome to try it as a learning exercise.

To specify an array or sequence of values in an answer, you must separate the values by a single space character (with no punctuation and with no leading or trailing whitespace). For example, if the question asks for the first ten powers of two (starting at 1), the only accepted answer is:

1 2 4 8 16 32 64 128 256 512

If you wish to discuss a particular question and answer in the forums, please post the entire question and answer, including the seed (which is used by the course staff to uniquely identify the question) and the explanation (which contains the correct answer).

☐ **In accordance with the Coursera Honor Code, I (Atul Gupta) certify that the answers here are my own work.**

Question 1

(seed = 885613)
Which of the following id[] array(s) could be the result of running the weighted quick union algorithm on a set of 10 items?

☐ 4 4 1 2 4 4 8 9 8 2

☐ 6 5 4 1 5 6 6 5 5 1

☐ 8 7 8 1 7 7 8 8 3 8

☐ 0 8 9 0 6 7 0 0 0 8

☐ 0 1 9 3 4 9 9 7 8 9

Question 2

(seed = 576529)
Suppose that you binary search for the key 15 in the following sorted array of size 15:

13 14 15 23 24 38 44 52 59 62 66 73 74 76 97

Give the sequence of keys in the array that are compared with 15.

Question 3

(seed = 676773)
Suppose that you time a program as a function of N and produce the following table.

```
       N   seconds
-------------------
     216      0.02
    1296      4.84
    7776   1254.72
```

Estimate the order of growth of the running time as a function of N. Assume that the running time obeys a power law T(N) ~ a N^b. For your answer, enter the constant b. Your answer will be marked as correct if it is within 1% of the target answer - we recommend using two digits after the decimal separator, e.g., 2.34.

Question 4

(seed = 983729)
What is the order of growth of the worst case running time of the following code fragment as a function of N?

```
int sum = 0;
for (int i = 1; i <= N*N*N; i++)
    for (int j = i+1; j <= N; j++)
        sum++;
```

- 1
- log N
- N^1/2
- N
- N log N
- N^2 log N
- N^3/2
- N^2
- N^5/2
- N^3
- N^4
- N^5
- N^6
- N^7

Question 5

(seed = 230905)
Consider an object of type GenericMysteryBox<Boolean> that stores N items of type Boolean.

```
    public class GenericMysteryBox<Item> {
        private int N;
        private Item[] items;

        ...
    }
```

Using the 64-bit memory cost model from the lecture, how many bytes does it use as a function of N? Include all memory referenced by the object and use tilde notation to simplify your answer. For example, enter ~1N if the number of bytes is 1N + 32.

Assume that the length of the array equals the number of items stored in the data structure.

Hint: An object of type Boolean uses 24 bytes.

Question 6

(seed = 133044)
Suppose that you have a data type for a sequence of N items and that it is represented internally using a doubly-linked list. Assume t
hat the data type is implemented in an efficient and natural manner given the specified representation.

Match up each of the following operations with their worst-case running time.
You may use each number once, more than once, or not at all.

___ return the last item in the sequence

___ insert a specified item immediately after the ith item in the sequence

___ is a specified item in the sequence?

___ insert a specified item at the beginning of the sequence

___ return the number of items in the sequence

___ remove and return the first item in the sequence

0. 1
1. log N
2. N
3. N log N
4. N^2

Question 7

(seed = 706579)
Give the array that results immediately after the 4-sorting phase of
Shellsort on following array:

21 86 43 97 12 80 66 58 17 46

Question 8

(seed = 436076)
Give the array that results immediately after the 7th calls to merge()
when bottom-up mergesorting the following array:

93 22 82 84 94 62 32 60 36 68

Question 9

(seed = 929334)
Give the array that results after applying quicksort partitioning to the following array:

30 68 34 48 44 65 31 83 26 47 22 84

Use the standard partitioning algorithm, in which the leftmost entry is the partitioning item.

Question 10

(seed = 956081)
Give the sequence of the 7 keys in the array that results after performing 3 successive delete-the-max operations on the following maximum-oriented binary heap of size 10:

97 96 84 54 93 33 72 40 49 25

Question 11

(seed = 933038)
Suppose that you have a priority queue containing N keys that is represented internally using a sorted array (where keys are ordered from smallest to biggest). Assume that the data type is implemented in an efficient and natural manner given the specified representation.

Match up each of the following operations with their amortized running times.
You may use each number once, more than once, or not at all.

___ return a maximum key	0. 1
___ insert an array of N specified keys	1. log N
___ delete and return a minimum key	2. N
___ iterate over the keys in ascending order	3. N log N
___ return a minimum key	4. N^2
___ delete and return a maximum key	

Question 12

(seed = 65907)
The column on the left contains the original input of 24 strings to be sorted or shuffled;
the column on the right contains the strings in sorted order; the other columns contain the contents at some intermediate step during one of the 9 algorithms listed below.

Match up each column with the corresponding sorting or shuffling algorithm from the list given below.
You should use each algorithm exactly once. That is, your answer should be a permutation of the 11 integers 0 to 10, starting with 0, ending with 10, and with each integer separated by a single space.

0	ibex	crow	bear	bear	calf	boar	boar	bull	boar	wren	bear
1	puma	puma	hake	boar	bear	deer	bull	crab	bull	worm	boar
2	seal	mole	frog	bull	hake	hare	calf	bear	crow	wasp	bull
3	mole	boar	dove	calf	frog	ibex	crab	boar	deer	wolf	calf
4	wren	worm	deer	crab	dove	mole	crow	calf	hare	sole	crab

```
 5    deer    bull    hare    crow    deer    puma    deer    crow    ibex    seal    crow
 6    hare    wren    boar    deer    hare    seal    duck    frog    mole    hare    deer
 7    boar    seal    bull    dove    boar    wren    hare    dove    puma    lion    dove
 8    bull    hare    crow    duck    bull    bull    ibex    duck    seal    slug    duck
 9    crow    deer    crab    frog    crow    calf    lion    deer    wasp    kiwi    frog
10    worm    wasp    duck    hake    crab    crab    mole    hake    worm    puma    hake
11    wasp    ibex    calf    hare    duck    crow    puma    hare    wren    deer    hare
12    calf    calf    ibex    mole    ibex    duck    seal    kiwi    calf    calf    ibex
13    duck    duck    wasp    seal    wasp    lion    wasp    ibex    crab    duck    kiwi
14    crab    crab    lion    wren    worm    wasp    worm    lion    dove    crab    lion
15    lion    lion    worm    lion    lion    worm    wren    seal    duck    boar    mole
16    dove    dove    slug    puma    wren    bear    dove    mole    lion    dove    puma
17    slug    slug    wolf    slug    slug    dove    slug    sole    slug    mole    seal
18    wolf    wolf    wren    wolf    wolf    frog    wolf    puma    bear    bull    slug
19    frog    frog    kiwi    wasp    mole    hake    frog    wasp    frog    frog    sole
20    kiwi    kiwi    sole    kiwi    kiwi    kiwi    kiwi    slug    hake    crow    wasp
21    sole    sole    mole    sole    sole    slug    sole    wren    kiwi    ibex    wolf
22    hake    hake    seal    worm    seal    sole    hake    wolf    sole    hake    worm
23    bear    bear    puma    ibex    puma    wolf    bear    worm    wolf    bear    wren


      ___     ___     ___     ___     ___     ___     ___     ___     ___     ___     ___

 0.  Original input
 1.  Selection sort
 2.  Insertion sort
 3.  Shellsort (with 3x + 1 increment sequence)
 4.  Mergesort (top-down)
 5.  Mergesort (bottom-up)
 6.  Quicksort (standard, no shuffle)
 7.  Quicksort (3-way, no shuffle)
 8.  Heapsort
 9.  Knuth shuffle
10.  Sorted
```

Question 13

```
(seed = 358948)
What is the order of growth of the expected running time when running the sorting algorithms
in the left-hand column on an input of size N whose keys are reverse sorted (with distinct keys)?

Assume that the sorting algorithms in the left-hand column are the pure, unoptimized verisons.
You may use each number once, more than once, or not at all.

___ heapsort                                0. 1

___ randomized quicksort (3-way)            1. log N

___ mergesort (top-down)                    2. N

___ mergesort (bottom-up)                   3. N log N

                                            4. N^2
```

Question 14

(seed = 966001)
Give the level order traversal of the BST that results after inserting
the following sequence of keys into an initially empty BST:

89 24 95 22 66 81 90 87 25 36

Question 15

(seed = 194047)
Consider the left-leaning red-black BST whose level order traversal is

18 13 67 11 17 50 84 49 58 42 (red links = 50 42)

What is the level order traversal of the red-black BST that results after
inserting the following sequence of keys:

10 85 40

Question 16

(seed = 720863)
Suppose that you insert the following sequence of points into an initially empty kd-tree.
Give the level order traversal of the resulting kd-tree.

A (0.53, 0.35)
B (0.79, 0.67)
C (0.35, 0.08)
D (0.77, 0.63)
E (0.20, 0.06)
F (0.81, 0.65)
G (0.29, 0.14)
H (0.84, 0.91)

Recall that our convention is to subdivide the region using the x-coordinate at even levels
(including the root) and using the y-coordinate at odd levels. Also, we use the left subtree
for points with smaller x- or y-coordinates.

Question 17

(seed = 560365)
For each quantity in the left column, give the best-matching description from the right column.
You may use each number once, more than once, or not at all.

___ Min height of a BST with N keys	0. 0
___ Min height of a left-leaning red-black BST with N keys	1. ~ 1
___ Min height of an (unweighted) quick union forest with N items	2. ~ 1/2 lg N
___ Max height of a (perfectly balanced) 4-way heap (each node has 4 children) with N keys	3. ~ log_3 N
___ Min height of a weighted quick union forest with N items	4. ~ ln N

___ Max height of a BST with N keys

5. ~ lg N

6. ~ 2 lg N

7. ~ 2 ln N

8. ~ N

Question 18

(seed = 165755)
Give the array that results after inserting the following sequence of 10 keys
into an initially empty linear probing hash table.

key	hash
I	3
J	4
G	1
K	5
S	3
L	6
B	6
T	4
P	0
W	7

Assume that the size of the hash table is 10 and that it does not grow or shrink.

Question 19

(seed = 157148)
Suppose that you have a data type for a set of N items (no duplicate keys) and that it is represented internally using an ordered arra
y (with the ith smallest key at array entry i). Assume that the data type is implemented in an efficient and natural manner given the
specified representation.

Match up each of the following operations with their amortized running times.
You may use each number once, more than once, or not at all.

___ remove a specified key from the set

___ minimum key

___ delete the minimum key

___ is a specified key in the set?

___ number of keys <= a specified key

___ kth largest key

0. 1

1. log N

2. N

3. N log N

4. N^2

Question 20

(seed = 821051)
You are applying for a job at a new software technology company. Your interviewer asks you
to identify the following tasks as either possible (with algorithms and data structures learned
in this course), impossible, or an open research problem.

You may use each number once, more than once, or not at all.

0. Possible

1. Impossible

2. Open

___ Implement a union-find data type so that all operations (except construction) take constant time in the worst case.

___ Given an array of N distinct numbers, determine whether there are three entries that sum to exactly 0 in N^1.5 time.

___ Prove that every compare-based sorting algorithm uses at least ~ 9/8 N lg N compares in the worst case.

___ Design a compare-based sorting algorithm that guarantees to sort any array of N comparable keys using ~ 9/10 N lg N compares in th
e worst case.

___ Find the smallest key greater than or equal to a given key in a left-leaning red-black BST in logarithmic time.

___ Find the kth largest key in a left-leaning red-black BST in logarithmic time.

___ Determine how many keys in an ordered array are less than a given key in logarithmic time.

___ Return a list of all intersections among a set of N orthogonal line segments in linearithmic time.

In accordance with the Coursera Honor Code, I (Atul Gupta) certify that the answers here are my own work.

Submit Answers Save Answers

You cannot submit your work until you agree to the Honor Code. Thanks!