Exam 1
CS-GY 6033 INET SPRING 2025
March 3rd 2025

Instructions:

Scheduling:

- The exam is runs from 6:00pm to 9:00pm on March 3rd 2025. The exam will be available to download from Gradescope at 5:50pm. The exam is to be completed by 9:00pm. Gradescope stops accepting all uploads at 9:15pm. There are absolutely no late uploads accepted by the system after that time. The time to complete the exam will depend on your preparation: a prepared student could finish it in less than two hours, an ill-prepared student might take up to three hours. It is your responsibility to allow time for scanning and uploading your exam within the deadlines.

Format:

- The written exam consists of a total of 100 possible points, but will be graded out of 90. The oral is worth an additional 10 points.
- You may write your solutions directly on the downloaded exam paper, *or* in your own format. You are responsible for providing clear and legible solutions to the problems. Your exam must be resubmitted into Gradescope electronically. Ensure that you know how to digitally scan any handwritten material. This is entirely the student's responsibility.

Questions during the exam:

- There is a ZOOM session for questions that will be open during the entire course of the exam, with MICROPHONES OFF. You may ask questions to the instructor with private chat during the exam. Any announcements made by the instructor during the exam will be made over ZOOM and also be email. **It is the student's responsibility to stay connected (either by ZOOM or email) during the exam.**

Rules:

- This exam is a **take-home exam**. You may use **only** the resources from the online class (course notes, video slides, problem solutions) and any type of calculator (although it is not needed). You may not copy or reference **any** textbook.
- Do not copy from personal notes, shared notes, or notes you have copied from any outside source.
- Your work must be entirely your own. It is **forbidden to discuss any work with *any* other person**. Furthermore, your work must be done without using internet searches (although this is completely unhelpful for this exam). Any breach of academic honesty will be handled in accordance with the *Student Code of Conduct*, (a copy of which is provided), and in this particular case, taken very seriously.
- You are asked to **read** the attached Student Code of Conduct Section III subsections A,B,C,D,E and **sign** below to acknowledge that you aware of the policy. Once signed, a copy of this page must be uploaded with your exam.

I acknowledge that my submitted Exam work is entirely my own. I have read and am in accordance with the Student Code of Conduct policy of NYU Tandon and fully accept the consequences of breaching the above instructions.

Name: _____

Signature: _____

.

Instructions:

Justify your answers to each question.

You may reference and use algorithms and results that we have seen in class.

Questions during the exam:

Join ZOOM meeting, link on Brightspace under ZOOM or Calendar

https://nyu.zoom.us/j/92575849617

Question 1: SHORT ANSWER! Each part is worth 2 points.

(a) Suppose an algorithm has a runtime given by the equation $T(n) = \sqrt{n^3 + 1} + n \log n$. Which of the following are true for $T(n)$?. You do not need to justify your answer:

$$O(n^2), O(n \log n), O(n^{1.5}), O(n), \Theta(n), \Theta(n^3), \Omega(n \log n), \Omega(n), \Omega(\log n)$$

(b) Suppose $T[0, 1, \ldots, 10]$ is a hash table where hashing is carried out by quadratic hashing with $h(k, i) = k^2 + 2i + 3i^2 \mod 11$. If we insert key k, is it possible that we need to evaluate $h(k, 12)$? Explain why.

(c) Let $T[0, 2, \ldots, 9]$ be a hash table of size 10, where hashing is carried out with double hashing where $h_1(k) = k \mod 10$ and $h_2(k) = k \mod 10$. Consider searching for the keys $k = 1, 2, 3, 4, 5$. For which of the five keys does the hash sequence **not** search the entire table? Explain why.

(d) Suppose we insert 100 keys into a hash table of size 25 where hashing is carried out with a uniform hash function, and collisions are resolved with chaining. What is the expected length of a chain?

(e) Suppose a recursive algorithm has runtime recurrence $T(n) = 4T(2n/3) + n^3$. What is the asymptotic runtime of the algorithm? Write your solution in big-Theta notation.

(f) Does the following procedure correctly sort the array $A[]$ indexed between s and f? If the procedure is incorrect, given an example of where it fails. Otherwise, explain in 1-2 sentences why it works. (*You may assume that insertion sort returns if array indices indicate no elements*)

```
MySort(A, s, f)
    if s < f
        q = round-down((s + f)/2)
        MySort(A, s, q)
        InsertionSort(A, q + 1, f)
        Merge(A, s, q, f)
```

(g) Suppose $T[]$ is a hash table of size 10. Currently there are 3 elements in the table. If we use a uniform hash function to insert a new key, what is the chance that there is a collision?

(h) Consider a max heap with 100 elements. If we are looking for the fourth largest element, which level could it be in? List all possible levels. Reminder that the root node is at the zero-th level.

(j) Consider the Select algorithm for finding the element of rank k in array A. What is the **best-case** runtime of this algorithm?

(k) Which of the following procedures have a best-case runtime of $O(n)$?

Bubble-sort, Partition, Randomized-Select, Select, HeapSort, QuickSort

Question 2

8 points

Rank the following functions in order (non-decreasing) of their asymptotic growth. Next to each function, write its big-Theta value, (ie. write the correct $\Theta(g(n))$. You are not required to prove the big-Theta value). *Reminder: you may assume* log *is base two.*

$$3^{\log_3 n}, \quad \log(2n^{50}), \quad \sqrt{\log n + 100^{100}}, \quad (\sqrt{n}+1)^2(\log n + 5), \quad \frac{1+\log n + n^2}{n + 6\log n},$$

$$\frac{1+20n^2}{n^2+6\log n}, \quad (n\log n + 2^{10})^2, \quad \log(5+2^n), \quad \frac{3^n+n}{n+2^n}, \quad 2^{3n+6}, \quad 3^{2n+6}, \quad n!, \quad n^n$$

Question 3

6 points

The following function is defined for $n > 2$:

$$f(n) = \frac{3n^3 + (\log n)^2}{n^2 - n} + 1000$$

is $\Theta(g(n))$ for the correct function $g(n)$. You must use the definitions from class, and *explicitly define* $n \geq k$

Question 4

6 points

The algorithm below takes as input an array $A[]$ indexed between s and f.

```
SlowSort(A, s, f)
    if s < f
        q =Random(s, f)
        SlowSort(A, s, q)
        SlowSort(A, q + 1, f)
        Merge(A, s, q, f)
```

Does the algorithm sort correctly? Explain why, or provide an example to show where it fails.

Write the best-case runtime recurrence for the procedure. Using this recurrence, what is the best-case runtime of the algorithm?

Write the worst-case runtime recurrence for the procedure. Using this recurrence, what is the worst-case runtime of the algorithm?

Question 5

4 points

Let $A = [3, 2, 4, 6, 1, 9, 8, 5, 7]$. Below are three 'shapshots' of intermediary stages of a sorting algorithm applied to A. For each sequence, determine *which* sorting algorithm produced that sequence. You must justify your selection (you do not need to explain why you eliminated the other options). You must select from: Selection sort, Insertion sort, QuickSort, Bubble Sort.

- $1, 2, 4, 6, 3, 9, 8, 5, 7$
- $2, 3, 4, 6, 1, 9, 8, 5, 7$
- $3, 2, 4, 6, 1, 5, 7, 9, 8$
- $2, 3, 4, 1, 6, 8, 5, 7, 9$

Question 6

9 points

The recursive procedure below takes as input an array A indexed between s and f:

```
TestRun(A, s, f)
  if f − s ≥ 4
      m = round-down((s + f)/2)
      q= round-down((s + m)/2)
      TestRun(A, s, q)
      TestRun(A, q + 1, m)
      SelectionSort(A, m + 1, f)
```

(a) 4 points Write the recurrence relation for the runtime.

(b) Use the recursion tree method to find a tight asymptotic upper bound (written in big-oh notation) for the algorithm.

(c) Use the substitution method to show that you get the same asymptotic bound as above.

Question 7

6 points

Draw the decision tree that models the execution of the partition algorithm on input $[a, b, c, d]$. Each variable represents a numerical value. For the execution, you do not need to pick a random pivot. Instead, use the last element of the subarray as the pivot. Justify why the height of the tree corresponds to the asymptotic runtime of Partition from class.

Question 8

6 points

Suppose $T[0, 1, 2, \ldots 12]$ is a hash table where collisions are resolved with quadratic hashing, using $h(k, i) = k + i + 4i^2 \mod 13$. Insert the following keys into the table, showing any work needed to resolve collisions: $3, 23, 17, 27, 9, 19, 6, 16, 26$

Question 9

5 points

Consider the following proposed *version* of binary search called NewSearch(A, s, f, k), which takes as input an array A indexed between s and f **sorted in increasing order**, and a key k. The procedure returns TRUE if the key k is in the array, and FALSE otherwise.

```
NewSearch(A, s, f, k)
    if s < f
        q = Random(s, f)
        if A[q] = k
            return TRUE
        else if A[q] < k
            return NewSearch(A, q + 1, f)
        else
            return NewSearch(A, s, q − 1)
    else if s = f
        if A[s] = k
            return TRUE
    return FALSE
```

What is the best-case number of comparisons carried out by this algorithm?
What is the worst-case number of comparisons carried out by this algorithm?
Be sure to justify your answers!
Note that for the above you are asked for the* ***exact number of comparisons**. You are not providing a solution in big-Oh notation.*

Question 10

6 points

Consider the following two algorithms for building a max-heap from elements in the array A, with $A.heapsize$ set to the number of elements in the array. The initial call to the first algorithm is **Make-Heap(A,A.heapsize)** and the initial call to the second algorithm is **Make-Heap2(A,1)**

```
Make-Heap(A, i)
    if i ≤ A.heapsize
        Bubble-up(A, i)
        Make-Heap(A, i − 1)
```

```
Make-Heap2(A, i)
    if i ≤ A.heapsize
        Bubble-up(A, i)
        Make-Heap2(A, i + 1)
```

For each algorithm, determine if the procedure correctly builds the heap. Justify your answers.

Question 11

6 points

Suppose A is a max-heap, where A.heapsize is defined as in class. Exactly one element in the heap was accidentally changed to a *smaller* value. However, you don't know *which* element was changed. Your job is to write a procedure that corrects the heap. This involves both finding an element that violates the heap property, and then repairing the heap as necessary. Call your **recursive** procedure FixHeap(A, i), where the initial call to the algorithm is FixHeap$(A, 1)$.

Justify the runtime of your algorithm as $O(n)$.

Question 12

6 points

- Let $A[]$ be an array of n natural numbers, written in binary, where each number is at most n.
- Let $B[]$ be an array of n fractions, where each fraction is of the form $\frac{1}{k}$ where k is a natural number less than 1000.

For each input array A and B, describe how to sort the input using radix sort. Explain any pre-processing steps needed, and justify the runtime.

Next, suppose the numbers in A were uniformly distributed. Explain how to use Bucket sort to sort the input in expected time $O(n)$

Question 13

6 points

Suppose you have an array containing n distinct numbers, where n is a multiple of 4. Describe an $O(n)$ algorithm that outputs the "*middle*" $n/2$ elements, that is, those that are within $n/4$ of the median. For example, for $n = 16$ in the example below, the middle $n/2$ elements are: $14, 6, 7, 10, 13, 15, 19, 20$.

$$4, 26, 23, 14, 3, 6, 7, 1, 2, 10, 13, 15, 24, 31, 20, 19$$

Question 14

6 points

Suppose Quicksort uses by default the last element in the array as the pivot. Execute quicksort on the array $A = [8, 1, 4, 6, 3, 2]$ using by default the last element of each subarray as the pivot.

You must:

-show the parameters of each recursive call

-show the *result* of each call to the in-place partitioning algorithm (you do not need to show the in-between steps of partition, just the final output.