ECE GY 9343 MIDTERM EXAM (2022 Spring)

Name: Minrui Li **NetID:** ml7136

Session(Circle one): Session A (Prof. Yong Liu) Session B (Prof. Pei Liu)

Answer ALL questions. Exam is closed book. No electronic aids. However, you are permitted two cheat sheets, two sides each sheet. Any content on the cheat sheet is permitted.

Multiple choice questions may have **multiple** correct answers. You will get **partial credits** if you only select a subset of correct answers, and will get zero point if you select one or more wrong answers.

Requirements for in-person students ONLY

Please answer all questions on the question book. You should have enough space. Since we scan all the submissions in a batch, DON'T write on the back of the pages and don't tear off any page.

Make sure you write your NetID on the bottom of each page (not your N number).

If you need extra scrape papers, we can give to you. However, it will not be graded.

Requirements for remote students ONLY

Each student is required to open Zoom and turn on their video camera, making sure the camera captures your hands and your computer screen/keyboard. The whole exam will be recorded. To clearly capture the video of exam taking, it is recommended that: A student can use an external webcam connected to the computer, or use another device (smartphone/tablet/laptop with power plugged in). Adjust the position of the camera so that it clearly captures the keyboard, screen and both hands. During the exam, you should keep your video on all the time. If there is anything wrong with your Zoom connection, please reconnect ASAP. If you cannot, please email ASAP, or call your proctor.

During the exam, if you need to use the restroom, please send a message using the chat function in Zoom, so we know you left.

Please use a separate page for each question. Clearly write the question numbers on top of the pages. When you submit, please order your pages by the question numbers.

Once exam is finished, please submit a single PDF on Gradescope, under the assignment named Midterm Exam. DEADLINE for submission is 15 minutes after the exam ended. Please remember to parse your uploaded PDF file. Before you leave the exam, its your responsibility to make sure all your answers are uploaded. If you have technical difficulty and cannot upload 10 minutes after exam ended, email a copy to your proctor and the professor.

1. **(20 points) True or False**

   (a) **T or F:** An array with the values of $[15, 9, 3, 6, 5, 4, 1]$ is a max heap.

   (b) **T or F:** If any one element in a max-heap with n nodes is changed, the heap property can be restored in $O(n)$ time;

   (c) **T or F:** It takes $\Theta(n)$ time to check if an array of length n is sorted or not;

   (d) **T or F:** Bubble-sort is stable;

   (e) **T or F:** Counting sort is NOT in-place;

   (f) **T or F:** If we are using the division method for our hashing function, a table size of 128 would be a good choice.

   (g) **T or F:** If chaining is used to handle collisions, the first element inserted to the hash table is always at the head of the chain;

   (h) **T or F:** Complexity to find the min or max node of a binary search tree is a constant. $O(h)$

   (i) **T or F:** $3^n = \Theta(2^n)$;

   (j) **T or F:** When input size is very large, a divide-and-conquer algorithm is always faster than an iterative algorithm that solves the same problem;

d

2. **(4 points)** Which of the following statements about hash tables are true?

   (a) Search for the maximum node always takes $\Theta(n)$;

   (b) When handling collisions using chaining, search for a non-existing node always takes $\Theta(n)$;

   (c) With universal hash functions, a random hash function is picked each time whenever a new item is inserted into the table;

   (d) Hash tables are preferred over direct-address tables due to possible reduction in storage requirement;

   (e) None of the above

b c e

3. **(4 points)** Given a max-heap with height 8 and distinct keys, the 3-rd largest element can be at height of:

   Height

   (a) 1;

   (b) 5;

   (c) 7;

   (d) 8;

   (e) 6;

b d

4. **(4 points)** If $f(n) = \Theta(g(n))$, and $g(n) = \omega(z(n))$, which of the following are true?

   (a) $f(n) = \Theta(z(n))$;

   (b) $f(n) = \Omega(g(n))$;

   (c) $g(n) = O(z(n))$;

   (d) $f(n) = \omega(z(n))$;

   (e) None of the above

c d

5. **(4 points)** Which of the following sorting algorithms run in worst-case time $O(nlogn)$?

   (a) QuickSort;

   (b) InsertionSort;

   (c) HeapSort;

   (d) MergeSort;

   (e) None of the above

NetID: ml7236

6. (**4 points**) Which of the following about binary search tree is true?

b d e

- ~~(a)~~ the maximum key is always on a leaf node;
- (b) in-order tree walk prints out all keys in sorted order;
- ~~(c)~~ binary search tree is a close-to-complete binary tree;
- (d) the worst-case search time in a binary search tree with n nodes is $\Theta(n)$;
- (e) for any key in the left subtree of a node z is less than or equal to any key in the right subtree of z;

7. (**6 points**) Illustrate the operation of insertion-sort on the array [71, 25, 40, 7, 60, 13, 20, 80]

$\boxed{71}$ 25 40 7 60 13 20 80

71 $\boxed{25}$ 40 7 60 13 20 80

25 71 $\boxed{40}$ 7 60 13 20 80

25 40 71 $\boxed{7}$ 60 13 20 80

7 25 40 71 $\boxed{60}$ 13 20 80

7 25 40 60 71 $\boxed{13}$ 20 80

7 13 25 40 60 71 $\boxed{20}$ 80

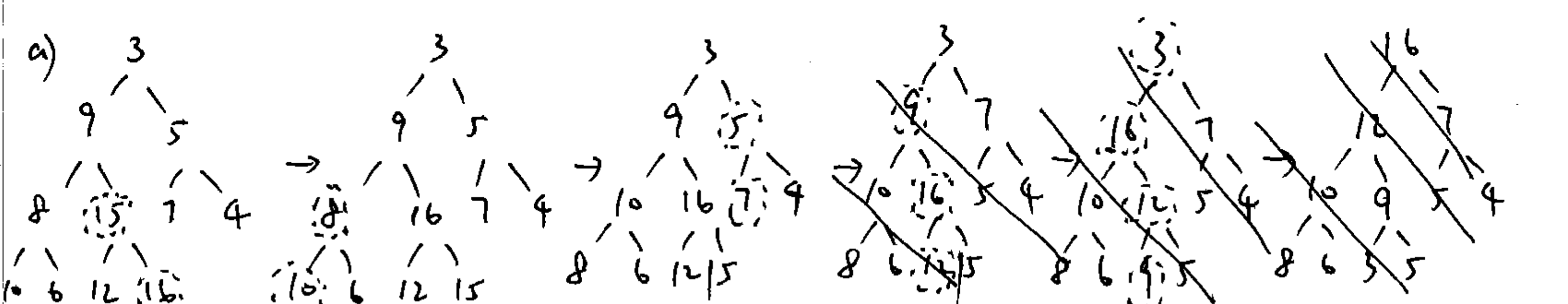
7 13 20 25 40 60 71 $\boxed{80}$

8. (**6 points**) Given an array of $[3, 9, 5, 8, 15, 7, 4, 10, 6, 12, 16]$,

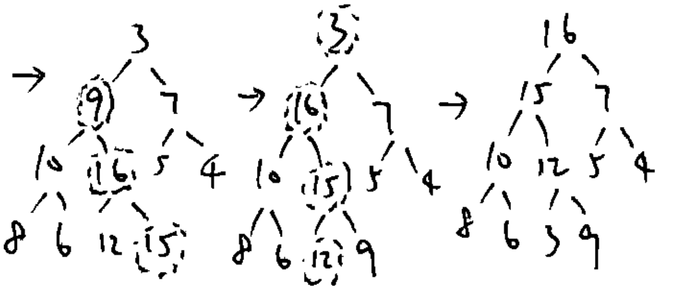
(a) (**3 points**) plot the initial max-heap built from the array;

(b) (**3 points**) plot the max-heap after the maximum is extracted, then a new number 11 is inserted.

a)

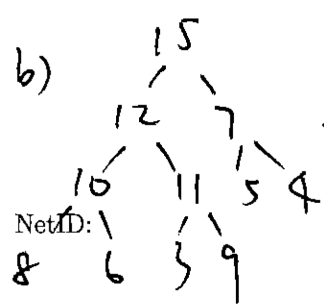


~~⇒ [16, 12, 7, 10, 9, 5, 4, 8, 6, 3, 5]~~



→ [16, 15, 7, 10, 12, 5, 4, 8, 6, 3, 9]

b)



→ [15, 12, 7, 10, 11, 5, 4, 8, 6, 3, 9]

NetID: ml7236

9. (**12 points**) Solve the following recurrences:

(a) (**4 points**) Use the iteration method to solve $T(n) = T(\frac{n}{2}) + 2T(\frac{n}{8}) + n^2$;

(b) (**4 points**) Use the substitution method to verify your solution for Question 9a.

(c) (**4 points**) Solve the recurrence $T(n) = 4T(\frac{n}{2}) + n^2 + n^{1.5}$.

a) $T(n)$ — — n^2

$T(n/2)$ $2T(n/8)$ — — $\frac{n^2}{4} + \frac{n^2}{32} = \frac{9}{32}n^2$

$T(n/4)$ $2T(\frac{n}{16})$ $2T(n/16)$ $4T(n/64)$ — $\frac{n^2}{16} + \frac{n^2}{64} + \frac{4n^2}{64^2} = \frac{81}{32^2}n^2$

$h_1 = \log_8 n$

$h_2 = \log_4 n$

$T(n) \geq n^2 + \frac{9}{32}n^2 + \frac{9^2}{32^2}n^2 + \cdots + (\frac{9}{32})^{h_1 - 1} n^2 \geq n^2$

$T(n) \leq n^2 + \frac{9}{32}n^2 + \cdots + (\frac{9}{32})^{h_2-1} n^2 \leq n^2 \cdot \sum_{i=0}^{\infty} (\frac{9}{32})^i = \frac{32}{23} n^2$

Therefore $n^2 \leq T(n) \leq \frac{32}{23} n^2$, $T(n) = \Theta(n^2)$

b) Suppose for all $k < n$, $T(k) = \Theta(k^2)$

Then $T(n) = T(\frac{n}{2}) + 2T(\frac{n}{8}) + n^2 \leq c\frac{n^2}{4} + 2 \cdot c\frac{n^2}{64} + n^2 = (\frac{9}{32}c + 1)n^2 = c'n^2$

$T(n) = T(\frac{n}{2}) + 2T(\frac{n}{8}) + n^2 \geq d \cdot \frac{n}{4} + 2 \cdot d \cdot \frac{n^2}{64} + n^2 = (\frac{9}{32}d + 1)n^2 = d'n^2$

Thus $d'n^2 \leq T(n) \leq c'n^2$, $T(n) = \Theta(n^2)$.

c) $T(n) = 4T(\frac{n}{2}) + n^2 + n^{1.5}$

By master's method, $n^{\log_2 4} = n^2 = \Theta(n^2 + n^{1.5})$

therefore, $T(n) = \Theta(n^2 \log n)$

NetID: ml7236

10. (**6 points**) **Prove** or **disprove** the following properties of asymptotic notations:

(a) (**3 points**) $n + \sqrt{n} = \omega(n)$, where $\omega(\cdot)$ stands for little-ω;

(b) (**3 points**) If $f(n) = \omega(g(n))$, then $g(n) = o(f(n))$, where $o(\cdot)$ stands for little-o.

~~a) False. $n+\sqrt{n} = \Theta(n) \neq \omega(n)$.~~
~~TRUE. Proof: Let $0 < c < 1$, $n+\sqrt{n} \geq c\,n$ for all $n > 0$~~.

b) TRUE, Proof:

$f(n) = \omega(g(n)) \rightarrow f(n) < c \cdot g(n)$ for some $c > 0$ and $n > n_0$

$\rightarrow g(n) > \frac{1}{c} f(n)$ for some $\frac{1}{c} > 0$ and $n > n_0$

$\rightarrow g(n) = o(f(n))$

a) False. $n+\sqrt{n} = \Theta(n) \neq \omega(n)$.

~~Let $c=2$, $n=1$, $n+\sqrt{n}=2$ #~~

~~$c \cdot n = 2$~~

~~$n+\sqrt{n} = c \cdot n$, Therefore,~~

Let $n+\sqrt{n} < c \cdot n$, then we get $c > \frac{1}{\sqrt{n}} + 1$

Which means there exists c s.t. $n+\sqrt{n} < c\,n$ for some n.

Therefore $n+\sqrt{n} = \omega(n)$ is false.

11. **(10 points)** The following is an iterative algorithm for binary search for a target value in an array nums with n numbers.

   (a) **(2 points)** analyze the worst-case complexity of the algorithm;

   (b) **(8 points)** prove the correctness of the algorithm using Loop-Invariant by clearly stating the Loop-Invariant statement, and proving initialization, maintenance, and termination steps.

```
int binarySearch(int nums[], int n, int target)
{  int low = 0, high = n - 1;
   while (low <= high)
   {  int mid = (low + high)/2;
      if (target == nums[mid]) {
         return mid;
      }
      else if (target < nums[mid]) {
         high = mid - 1;
      }
      else {
         low = mid + 1;
      }
   }
   return -1;
}
```

Termination: When low == high, we have reduced the size into 1. If this element is the target element, ~~we~~ the return the current position. Otherwise, return -1, because the target value is not contained in this input array nums.

a) The worst case will happen when we are trying to find the first or last element is the array.

In this case, we will keep doing mid = (low + high)/2 util we reach the first or last position. The total number of this operation is lgn, where n is the length of array.

Thus the time complexity of the worst-case is $T(n) = \Theta(\lg n)$.

b) Initialization: The initial Loop-Invariant is: at the very beginning, the target value must be contained in the original array.

Maintenance: At each iteration, we compare the value at position mid ~~if nums~~ with the target. If target == nums[mid], the we are done. If target < nums[mid], the target value must be located at the left side of position mid. If target > nums[mid], the target value must be located at the right side of position mid. In each case, we continue to find the target value at the subarray which ~~correct~~ could contain the target.

NetID: ml7236

Termination: ~~when low == high, we have reduced the size~~ ~~as~~ into ~~1, if this element is the target, we return the current position. Otherwise~~

12. **(9 points)** In Randomized Quicksort (with Lomuto's partition) for an array of n distinct numbers,

(a) **(4 points)** after the first call of randomized-partition, what is the probability that the i-th ranked number is in the left partition, for any $1 \le i \le n$?

(b) **(5 points)** suppose the second partition call is for the right partition resulted from the first partition call, what is the probability that the i-th ranked number is in the same partition as the $(i+k)$-th ranked number after the first two partition calls?

a) The ith ranked number is in the left partition means the ~~past~~ rank of pivot is greater than i. There are $n-i$ elements with the rank higher than i, each of them might be picked as the pivot with the probability $\frac{1}{n}$. Thus the probability that the ith ranked number is in the left partition $Pr = \frac{n-i}{n}$.

b) A = the ith ranked number and the $(i+k)$th ranked number ~~are~~ are at the left partition after first partition call.

B = ith and $(i+k)$th number are in the left partition of the right partition resulted from the first partition

C = ith and $(i+k)$th number are in the right partition of the right partition resulted from the first partition

$$P(A) = \frac{n-i-k}{n}, \quad P(B) = \sum_{p=0}^{i-1} \frac{i-1}{n} \cdot \frac{i-1-p}{n-p}, \quad P(C) = \sum_{p=0}^{i-1} \frac{i-1}{n} \frac{n-i-k}{n-p}$$

$$\cancel{P(B) = \left(1-\frac{n-i-k}{n}\right) \cdot \frac{i-1}{n} \cdot \frac{i-1}{n-1}} \quad \cancel{\frac{1}{i-1} \cdot \frac{1}{i-2} \cdot \frac{1}{2}} \cdot \cancel{\left(\frac{i-1}{2}\right)} \sum_{p=0}^{i-1} \cancel{\frac{1}{i-1} \cdot \frac{1}{i-p-1}}$$

$$\cancel{P(C) = \left(1-\frac{n-i-k}{n}\right) \cdot \frac{1}{n-i-k}} \quad \cancel{\frac{i-1}{n}} \cdot \cancel{\frac{1}{i-1}\ \frac{1}{n-p}}$$

Thus $Pr = P(A) + P(B) + P(C)$ $\cancel{= \frac{n-i-k}{n} + \frac{\binom{i-1}{2}}{2(i-1)(i-1)} + \frac{i-1}{n(n-i-k)}}$

$$= \frac{n-i-k}{n} + \sum_{p=0}^{i-1} \frac{i-1}{n} \cdot \frac{n-1-p-k}{n-p} \quad \cancel{\sum_{p=0}^{i-1} \frac{1}{i-1} \cdot \frac{1}{i-p-1}}$$

13. (**11 points**) We have a hash table with m slots and collisions are resolved by chaining. Suppose n distinct keys are inserted into the table. Each key is equally likely to be hashed to each slot.

   (a) (**5 points**) Let X_i be the random variable of the position of the i-th ($1 \leq i \leq n$) inserted key in its chain after all keys have been inserted (counting the position from the head of the chain), what is the probability mass function of X_i, i.e., calculate $P(X_i = k)$, $\forall k \geq 0$?

   (b) (**3 points**) What is the expected value of X_i?

   (c) (**3 points**) What is the expected number of elements examined when searching for a randomly selected key?

a) X_i = the amount of keys that are inserted into the same chain after ith key.

$$P(X_i = k) = \binom{n-i}{k}\left(\frac{1}{m}\right)^k\left(1-\frac{1}{m}\right)^{n-i-k}$$

b) X_i is a binomial random variable.

$$E(X_i) = \frac{n-i}{m}$$

c) $$E(\#\text{ of examined}) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{n-i}{m}+1\right)$$

$$~~= \frac{1}{nm}\sum_{i=1}^{n}(n-~~$$

$$= 1 + \frac{1}{nm}\sum_{i=1}^{n}(n-i)$$

$$= 1 + \frac{n-1}{2m}$$

This page intentionally is left blank. You can write answers here if you used up space in the front.
Don't tear off this page.

If you use up the space under any particular problem, you can write your answer here. On the page of the problem, tell us part of your work is here and write down the page number of this page and. Don't tear off any pages.

If you use up the space under any particular problem, you can write your answer here. On the page of the problem, tell us part of your work is here and write down the page number of this page and. Don't tear off any pages.