Exam 2 CS-GY 6033 INET Spring 2025
April 14th 2025

Instructions:

Scheduling:

- The exam runs from 6pm to 9pm Eastern time, on April 14th 2025. Your exam is run through Gradescope, and will available to download at 5:50pm. The exam is to be completed by 9:00pm. Gradescope stops accepting 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 uploading your exam.

Format:

- The written exam consists of a total of 100 possible points, and will be graded out of 90 points. Your oral quiz counts for 10 points, making a total of 100 points for exam 2.
- **Submission Penalty: if you do not submit your exam on Gradescope, or do not properly assign your pages, you receive a deduction of 3 points on your exam.**
- 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 quickly upload any handwritten material. This is entirely the student's responsibility. You may assign your pages after the deadline.

Questions during the exam:

- There is a ZOOM session for questions that will be open during the entire course of the exam (microphones OFF). You may ask questions 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 (any material on NYU classes for this course) and any type of calculator (although it is not needed).
- 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

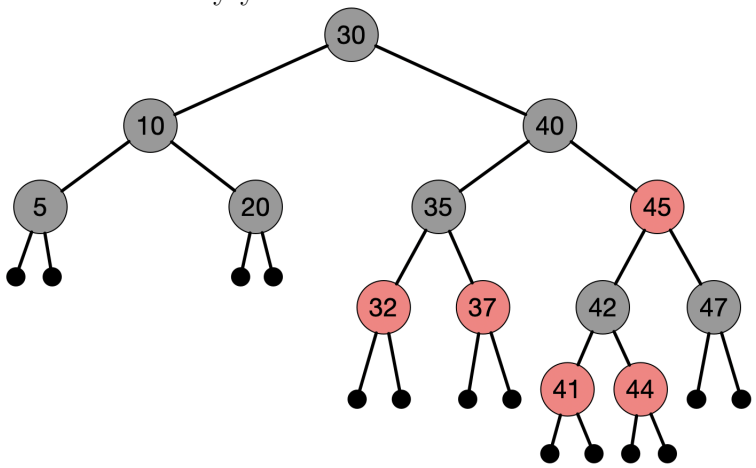
Read the following questions carefully. Recall that you may use algorithms from class. For example, you do not need to re-describe from scratch algorithms like inserting into a BST, Inorder Traversal, or using the Rank Algorithm.

You should aim to complete approximately **30 points** per hour.

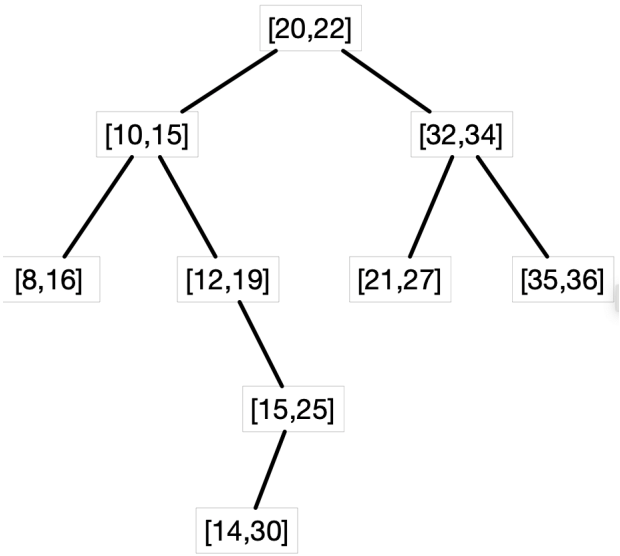
ZOOM LINK: https://nyu.zoom.us/j/93573710304
MICROPHONES OFF!

Question 1: SHORT ANSWER!

(a) 2 The drawing below represents a valid red-black tree. Does the insertion of 43 cause the black-height to increase? Justify your answer in one sentece.



(b) 2 points An interval tree is drawn below. Suppose we carry out the INTERVAL-SEARCH(i) algorithm from class, where $i = [16, 25]$. Show which node is returned.



(c) 2 points A red-black tree has 24 nodes. What is the largest possible black-height? What is the smallest possible black-height?

(d) 2 points. Can a BST have more than one pre-order traversal? Or is the pre-order traversal unique to a BST. Justify your answer.

(e) 2 points Let T be a BST augmented with subtree sizes, and let x be a reference to a node in the tree. Which of the following calls is guaranteed to return a reference to the successor of x? Select all that apply.

1. $k =$ BST-rank(T, x). Return x.right
2. Return FindMin(x.right)
3. Return FindMin(x.left)
4. $k =$ BST-rank(T, x). Return BST-Select$(T, k + 1)$
5. Return x.parent
6. Successor(T, x)

(f) 2 points Below is a completed table to the DP solution for WeightSelect. Recall that WeightSelect is the problem of determining if there are a set of elements that can be selected in such a way that the total weight is exactly equal to the target weight.

Using the information in the table, determine the following:

- the weight of item $i = 4$. In other words, what is $w[4]$?
- the weight of item $i = 2$. In other words, what is $w[2]$?

Weight index \ Target Weight	0	1	2	3	4	5	6
0	T	F	F	F	F	F	F
1	T	F	F	F	T	F	F
2	T	F	T	F	T	F	T
3	T	F	T	F	T	T	T
4	T	T	T	T	T	T	T

(g) 2 points Below is a completed table to the DP solution for MaxValueSet Recall that MaxValueSet is the problem of determining the maximum possible value of a selected set of items, whose total weight is at most T.

Using the information in the table, determine the following:

- the value of item $i = 4$. In other words, what is $v[4]$?
- the weight of item $i = 4$. In other words, what is $w[4]$?

Weight index \ Target Weight	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	4	4	4
2	0	0	1	1	4	4	5
3	0	0	1	1	4	4	5
4	0	6	6	7	7	10	10

(h) 2 points. Below is the output of the rod-cutting problem, where $n = 15$. Fill in the entry for $r[8]$ and $s[8]$.

$$p_1 = 2, p_2 = 5, p_3 = 7, p_4 = 8, p_5 = 6, p_6 = 20, p_7 = 10, p_8 = 15$$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
r[]	0	2	5	7	10	12	20	22								

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s[]	0	1	2	2	2	1	6	1								

(j) 2 points. Below is the output of the rod-cutting problem, where $n = 15$. What are the piece sizes that cut the rod optimally?

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
r[]	0	2	5	7	10	12	15	17	20	22	25	30	32	37	39	42

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s[]	0	1	2	2	2	1	2	1	2	1	2	11	1	2	1	2

(k) 2 points Below is the output of the longest common subsequence problem. Determine which of the following are true? (Select all that apply)

String A \ String B	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1
2	0	0	0	0	0	1	1
3	0	1	1	1	1	1	1
4	0	1	1	1	2	2	2
5	0	1	1	1	2	2	2
6	0	1	1	2	2	2	2

1. A[1] = B[1]
2. A[1] = B[6]
3. A[3] = B[6]
4. A[6] = B[3]

Question 2

4 points

Let T be the root of a red-black tree. Write the pseudo-code for a procedure called CountRed(T) which returns the number of red-colored nodes in the tree. Justify the runtime of $O(n)$.

Question 3

Suppose we are given a set of n points where each point is of the form (x, y), which represents the coordinates of a point in the xy plane. A search tree can be built using these points in the following way:

- at the *even* levels, (levels 0, 2, etc), the key used by the search tree is the *xcoord* coordinate.
- at the *odd* levels, (levels 1, 3, etc), the key used by the search tree is the *ycoord* coordinate.

(a) 2 points Using the above definition, show the tree that results by inserting the following elements one at a time into an initially empty tree:

$$(3, 5), (9, 5), (1, 3), (12, 4), (8, 6), (2, 1), (2, 5)$$

(b) 5 points The node objects of this tree have attributes *node.xcoord* and *node.ycoord*. Write the pseudo-code for an algorithm that inserts new node x into the search tree. Your procedure takes as input a reference T to the root of the tree, and a reference x to the new node. Your procedure must return a reference to the new tree after x is successfully inserted.

(c) 5 points Write the pseudo-code for an algorithm that returns a reference to the node with the minimum x coordinate. For example: in the above tree, the return value should be node $(1, 3)$.

Question 4

There are n athletes who compete in the 100m final at the summer olympics.

Let T be a binary search tree that stores information on the athletes finish times. Each tree node x contains the following attributes:

x.time: Finish time of the athlete, used as the *key* of the BST

x.totaltime The total of all finish times of athletes in the subtree rooted at x

x.age: Age of athlete x

x.maxage The maximum age out of all athletes in the subtree rooted at x.

x.mintme: Best finish time out of all athletes in the subtree rooted at x. Note that "best time" refers to the minimum time.

x.size:. Number of nodes in the subtree rooted at x

(a) 2 points Describe how to implement the above BST such that inserts and deletes can be carried out in time $O(\log n)$. Describe the runtime of building such a tree.

(b) 5 points Update the TreeInsert(T, x) procedure from class so that it inserts new tree node x into the tree T. Recall that the key used in the BST property is $x.time$. Ensure that your insertion properly assigns and updates all attributes: totaltime, age, maxage, mintime, size. As a reference, the Tree-Insert code from class is copied on the last page.

(c) 3 points Write the pseudo-code for a recursive algorithm called FastestTime(T), which returns a reference to the athlete with the fastest time. Justify the runtime of $O(\log n)$

(d) 6 points Write the pseudo-code for a recursive algorithm called MaxAge(T, k), which returns the **maximum age** out of all athletes with a finish time under k. Justify the runtime of $O(\log n)$.

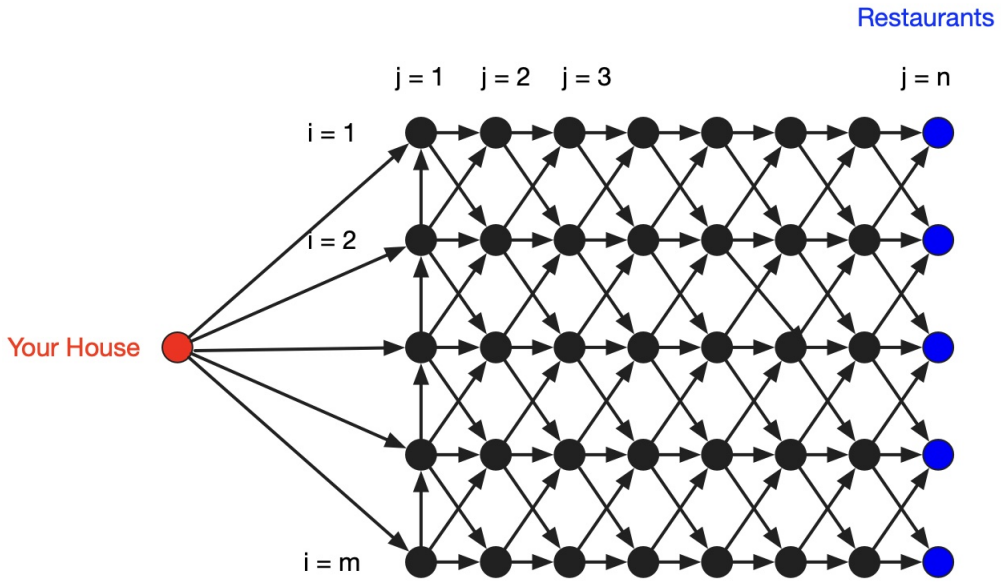
(e) 4 points Write the pseudo-code for an algorithm that returns the number of athletes with a finish time over k. Call your procedure AthletesOverTime(T, k). Justify the runtime of $O(\log n)$

(f) 6 points Write the pseudo-code for the procedure that finds the **average** finish time out of all athletes with a finish time over k. You may need to call more than one procedure. Justify the runtime of $O(\log n)$.

Question 5:

12 points

The figure below represents a city map. Each dot is a traffic light, and each directed line represents a one-way road. There are m rows of lights, and n columns. The red dot (left-most) is your house, and the blue dots (last column) are restaurants in the city. The blue restaurants are not traffic lights! Your goal is to leave your house and get to a restaurant as quickly as possible. However, each traffic light has an expected delay. The goal is to find the route from your house, to *any* restaurant, in the shortest expected time. The input to the problem is in the array $D[1 \ldots m, 1 \ldots n]$ where $D[i, j]$ represents the expected delay at light (i, j). Note that $D[i, n] = 0$ for the last column, because there is no traffic light delay at the restaurant.



Your Job:

Provide a DP solution for the problem of finding the least expected time it will take you to leave your house, and get to a restaurant.

You are required to use the following DP table: $T[1 \ldots m, 1 \ldots n]$ where $T[i, j]$ represents the minimum expected time it takes to leave your house and go to location (i, j).

Your solution must include explanations of the initialisation, justify how you fill in the entries, provide the pseudo-code, clearly show which value is returned, and justify the runtime.

Question 6

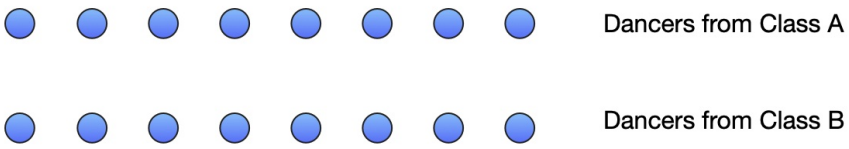
6 points

A country dance is being organised at your school. There are n students in class A and m students in class B. Dancers from these classes will perform in a line dance. Dancers will be arranged in rows, and stand directly across from their partner.

The rules are as follows:

- Dancers from class A must select a partner from class B of the **same age**
- In the final row dance formation, dancers from class A will form one row, dancers from class B will form the second row, where each dancer is standing directly across from their partner
- The dancers in row A must be arranged in order of **increasing height**. Similarly, dancers in row B must be arranged in order of **increasing height**

An example of the formation is shown below:



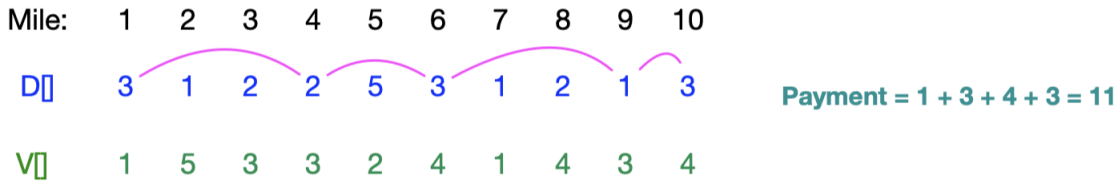
Note that because of these restrictions, not all students will be able to participate in the dance! The goal is to determine the maximum number of couples that can perform.

Describe how to solve this problem using a DP approach. You must either design your own solution, or describe how to solve the problem using material from class.

Question 7

20 points

Consider another hiking trail that goes from mile marker 1 to mile marker n. At each marker, there is again ONE rock. Each rock has an associated **distance** and **value**. For each rock that is picked up and carried its required distance, the hiker receives a payment which is equal the value of that rock. The input to the problem is $D[1, 2, \ldots, n]$ where $D[i]$ is the **distance** that we much carry rock i, and $V[1, 2, \ldots, n]$, where $V[i]$ is the value of rock i. For example, if $D[1] = 3$, the rock at marker 1 must be carried at least 4 miles, and in that case the hiker would receive a payment of $V[1]$ dollars. At each mile marker, we have the choice of picking up the rock, or continue walking with no rock. We cannot carry more than one rock at a time, but we are allowed to drop a rock, and pick a new one up at the same mile. Any rock that we pick up MUST be carried the minimum number of miles. In the example below, the rock pick ups and drop offs show that we are able to receive a payment of 11. This is not necessarily the optimal solution for the example below!



Part a:

Write a DP solution that solves the problem of finding the **maximum payment** the hiker can receive by picking up and dropping off rocks along his journey. Be sure to include a properly defined DP table, describe the initialisation, how the table is filled up, the pseudo-code, and the runtime.

Part b:

Update the above solution so that you also print out the optimal solution: this consists of the the mile markers where you pick up rocks.

REFERNCE FROM CLASS

TREE-INSERT(T,z)

```
if T = NIL
   return z
else x = T
While x ≠ NIL
    y = x
    if z.key < x.key
        x = x.left
    else x = x.right
z.parent = y
if z.key < y.key
    y.left = z
else y.right = z
return T
```