CS211FZ (Algorithms & Data Structures 2)

Assignment 5

Chris Roadknight

Chris.roadknight@mu.ie

Released date: Sunday, 8th June 2025

Deadline: Friday, 13th June 2025, at 18:00 Beijing time

This is an open-book, graded assignment. You may only use the module's slides and textbooks to help with the Assignment. Please cite all references as comments in your submissions. You cannot directly reuse a solution from online sources. You must not engage with another student, in person or electronically (via phone, social media, etc.), to secure assistance with this Assignment. If you do so (even for only one of the questions), you will receive an automatic failure (0%), and it will also be reported to the Executive Vice-Dean of MIEC and/or Maynooth University Plagiarism board. We will perform similarity checks on submitted assignments to check for collaborative efforts. The lecturer reserves the right to interview you about your submission in special cases.

Question 5.1 [10 marks]

Create a Red-Black Tree. Start with an empty tree and insert the keys 1, 2, 3, 4, 5 in order. After each insertion, they should draw the tree structure, indicating whether each node is red or black, and ensure the tree satisfies all red-black tree properties

Question 5.2 [15 marks]

Create an AVL tree. Start with an empty tree and insert the keys 7, 6, 5, 4, 3, 2, 1 in order. After each insertion, they should draw the tree structure, indicating the balance factor of each node, and ensuring the tree satisfies all AVL properties

Question 5.3 [10 marks]

You are given an initial population of 5 individuals, each represented as a 5-bit binary string. These strings represent numbers in binary form, and the fitness of each individual is calculated as Fi = 100/n

where (n) is the decimal equivalent of the binary string.

Here is the initial population:

String No.	String	n (decimal)	Fi =100/n
1	10111	23	4.35
2	00111	7	14.29
3	11000	24	4.17
4	01010	10	10.00
5	11101	29	3.45

- Selection: Use the roulette wheel method to select two parents for crossover.
  - Calculate the total fitness:

Total Fitness = sum Fi

.

  - Calculate the probability for each individual:

Probability = Fi/{Total Fitness}

.

  - Calculate cumulative probabilities and fill in the table below:

String No.	Fi	Probability	Cumulative Probability
1	4.35		
2	14.29		
3	4.17		
4	10.00		
5	3.45		

  - Generate a mating pool of 5 individuals (with replacement) using roulette wheel selection: for each, generate a random number between 0 and 1 and choose the individual whose cumulative probability interval contains it.
- Crossover: Randomly pair the mating pool individuals (e.g., pair 1 with 2, 3 with 4, 5 with 1). Perform single-point crossover with probability 0.5:
  - If crossover occurs, choose a random point between 1 and 4, swap segments after this point to create offspring.
  - If not, offspring are copies of parents. Create a new population of 5.
- Mutation: For each bit in the new population (25 bits total), with probability 0.02, flip the bit (0 to 1 or 1 to 0).
- Evaluation: Calculate (n) and

Fi = 100/n

for each new individual. Compare total and average fitness with the initial population.

Questions:

- **What is the total fitness of the initial population?**
- **What are the probabilities and cumulative probabilities?**
- **List the mating pool individuals.**
- **Show crossover operations and the new population before mutation.**
- **Show the population after mutation.**
- **Compare fitness: has the average improved?**

Question 5.4 [10 Marks]

Determine, algorithmically, a LCS of $\langle 1, 0, 0, 1, 0, 1, 0, 1\rangle$ and $\langle 0, 1, 0, 1, 1, 0, 1, 1, 0\rangle$.

Question 5.5 [15 marks]

Give a dynamic-programming solution to the 0-1 knapsack problem that runs in $O(n\ W)$ time, where n is the number of items and W is the maximum weight of items that the thief can put in the knapsack.

Question 5.6. [40 Marks]

This is a java programming question, you must demonstrate to your TA some progress on this question before you leave the lab.

Develop java software that answers question 5.3

Important submission details

Please indicate the Operating System (Linux/Windows/MacOS/Online), IDE (e.g. Eclipse, Visual Studio Code), and Java SDK version used for testing in your submission. If you use an online IDE, please specify the IDE (http://repl.it) and provide a link where possible.

All work must be submitted via Moodle (see "Assignments" section for submission). Work submitted via other means will not be accepted unless you have prior arrangements with the lecturer. All work MUST be submitted by the due date deadline. Late submissions will not be accepted.

Your submission should be one single PDF file, also including your Java codes. Submitting in any other format cannot be accepted.

Note: You should type your answers in a text editor like Microsoft Word, and then convert it to PDF. You MUST NOT take pictures from answers and then make a PDF from the pictures.