Quiz DP Practice with Answer

Quiz Submission Instructions:

During the quiz, you are permitted to have only two windows open on your computer:

- This PDF file containing the quiz questions.
- A code editor of your choice for writing down your answers. (such as VS Code)

Upon completion of the quiz, please follow these steps to submit your answers:

- Compile all your answers into the provided Java file (Quiz_Graph.java).
- Name the file using the following format: Graph_FirstName_LastName.java (replace FirstName and LastName with your actual first and last names).
- Email the file to the professor using the provided email address, x.qu@gwu.edu .

Important:

- Ensure that you adhere strictly to the allowed resources during the exam to maintain academic integrity.
- Verify that your answers are correctly saved in the JAVA file before submission.

Multiple Choice Questions:

1. What is Dynamic Programming?

A) A method to solve problems by breaking them into smaller problems and solving each just once.

B) A programming language.

C) A way to solve problems quickly without considering their subproblems.

D) A method that solves all problems without using recursion or iteration.

Correct Answer: A

2. Which of the following problems is best suited for a Dynamic Programming approach?

A) Finding the maximum value in an unsorted array.

B) Calculating the nth Fibonacci number.

C) Checking if a number is prime.

D) Converting a lowercase string to uppercase.

Correct Answer: B

3. What is Memoization?

A) A technique to remember solutions to subproblems in a recursive algorithm.

B) A spell-check feature in programming editors.

C) A memory management technique.

D) A debugging technique for recursive functions.

Correct Answer: A

4. Which approach starts solving the problem from the simplest possible subproblem and works its way up to the original problem?

A) Greedy Approach

B) Divide and Conquer

C) Bottom-Up Approach

D) Top-Down Approach

Correct Answer: C

5. In the context of Dynamic Programming, what is a 'state'?

A) The condition of the programming environment.

B) A temporary variable used in iterative solutions.

C) A unique configuration that represents a subproblem.

D) The current value of the recursive function's argument.

Correct Answer: C

6. Which of the following is NOT a characteristic of Dynamic Programming?

A) Overlapping subproblems

B) Optimal substructure

C) No need for storing intermediate results

D) Utilization of past knowledge to make solving future subproblems easier

Correct Answer: C

7. The process of solving each subproblem just once and storing its result is known as:

A) Storing

B) Memoization

C) Caching

D) Bookmarking

Correct Answer: B

Short Answer Questions:

1. Explain the difference between the Top-Down and Bottom-Up approaches in Dynamic Programming.

The Top-Down approach in Dynamic Programming starts with the main problem and breaks it down into smaller subproblems recursively, solving each subproblem only once and storing its result (memoization) to avoid redundant computations. This approach is intuitive as it mirrors the problem's natural recursive structure but may involve more recursive overhead.

On the other hand, the Bottom-Up approach begins with the simplest subproblems and iteratively combines their solutions to solve more complex subproblems, eventually arriving at the solution to the original problem. This approach typically involves filling up a table based on the dependencies of subproblems and often has better space and time efficiency due to the iterative nature and elimination of recursive call overhead.

2. How does Memoization contribute to the efficiency of Dynamic Programming solutions?

Memoization is a technique used in Dynamic Programming to store the results of expensive function calls and reuse those results when the same inputs occur again. This avoids the repetitive computation of the same subproblems, significantly reducing the time complexity of algorithms that involve many overlapping subproblems, such as those encountered in Dynamic Programming. By remembering previously computed values, memoization transforms exponential-time recursive algorithms into polynomial-time algorithms for many problems, thereby greatly enhancing their efficiency.

3. Describe how Pascal's Triangle can be used to understand the concept of Dynamic Programming.

Pascal's Triangle is an excellent example to illustrate the concept of Dynamic Programming due to its inherent structure of overlapping subproblems and optimal substructure. Each element in the triangle is the sum of the two directly above it, demonstrating how a complex problem (calculating a value in the triangle) can be broken down into simpler subproblems (the values above it).

Using Pascal's Triangle to calculate coefficients in binomial expansions showcases Dynamic Programming principles. For example, to calculate a specific value, one could use a top-down approach with memoization or a bottom-up approach to build the triangle iteratively. This highlights the efficiency gains from not recalculating values and the importance of solving subproblems to construct the solution to a larger problem, both key concepts in Dynamic Programming.

Coding Questions:

Coding Question:

We provide a skeleton of the code with comments indicating where students should add their code.

Top-Down Approach to Fibonacci Sequence: Write a Java function that uses a top-down approach with memoization to calculate the nth Fibonacci number.

Bottom-Up Approach to Fibonacci Sequence: Write a Java function that employs a bottom-up approach to compute the nth Fibonacci number.

The Code file is:

Quiz_dp.java