CS 5114 Theory of Algorithms, Spring 2024
Project 1: Due by 19 Feb. 2022, 11:59PM

I pledge that this test/assignment has been completed in compliance with the Graduate Honor Code and that I have neither given nor received any unauthorized aid on this test/assignment.

Name (Print): _____

Signed: _____

This project aims to enhance the understanding of the dynamic programming (DP) in solving optimization problems. A project will be graded based on the following criteria.

1. (10%) Select one optimization problem you want to solve using DP and give a specific problem description. For example, at least you should describe input parameters, an objective (i.e., minimization or maximization), and expected outputs.

2. (25%) Discuss why the selected optimization problem with a sufficiently larger size of the problem (e.g., $n \gg 100$) can be efficiently solved using dynamic programming, compared to other techniques, such as recursive algorithm. Discuss the benefit of using DP in the problem you selected. Follow the 4 steps we learned in the class to describe your approach.

3. (30%) Implement your selected problem using both DP (both top-down and bottom-up approaches) and recursive algorithm. Submit the zip file including your source code and README file including how to run your executable file. The input should be key design parameters to significantly impact an optimal value and optimal solution(s). The program should be able to print out the outputs that are the optimal value and optimal solution(s). Report the pseudo-code of the algorithms and example solution (optimal value/solution) with a case study based on specific input values. You may use graphs/figures to describe your example solutions.

4. (20%) Based on the results obtained from #3, compare running time (i.e., O, Ω, and Θ) of your DP algorithms (both top-down and bottom-up) with a recursive solution (i.e., compare three algorithms). Discuss how much the DP technique outperforms the recursive solution.

5. (15%) Identify/discuss additional key design parameter(s) or input values that may significantly affect optimal value/solutions (i.e., sensitivity analysis with respect to some key design parameters).

6. **Submission Format**:

   - Font 11; single-spaced
   - Do not exceed 5 pages in total
   - Submit the report in .pdf
   - Make a single .zip file including (1) Report; (2) Source codes files under a subfolder named 'source codes'; and (3) README file describing how to run your code.

7. **Each student will give a project presentation during class time on 2/20/2024. The project and its presentation should be submitted separately on the corresponding submission sites.**

Rubrics for P1 Report: Grading Criteria

(Include the pledge statement and sign on it at the beginning of your project report.)

No.	Total Score	Required Contents	Comments
1	10%	Explained clearly the following: • Input parameter description (3%) • Objective description (minimization or maximization) (4%) • Expected output description (3%)	
2	25%	The following criteria are sufficiently met: • DP is a right choice to solve a given optimization problem. (5%) • The key benefit of using DP is clearly stated. (5%) • The problem solved is challenging enough with a sufficiently large problem size. (15%)	
3	30%	Provided working implementation and algorithm descriptions with: • Top-down approach (7%) • Bottom-up approach (7%) • Recursive approach (7%) • Pseudo-code for all algorithms (5%) • Example solutions (4%)	
4	20%	Clearly discussed the following: • Running times of three algorithms (two DPs and recursive algorithm) in Big-O, Ω, and Θ (15%) • Performance comparison of the three algorithms (5%)	
5	15%	The following criteria are sufficiently met: • Conducted meaningful sensitivity analysis by varying the values of the key design parameters (10%) – (one sensitivity analysis: 5%; 2+ sensitivity analysis: 10%) • Clarified the reasons behind the observed results under the sensitivity analysis. (5%)	

Late penalty: 10% off per day (Any minute later after the deadline will be counted as one day late, e.g., 1 day late for the submission on 12:00am on the next day).

Rubrics for P1 Presentation: Grading Criteria

No.	Total Score	Task	Comments
1	10%	Explained the problem statement clearly including input parameters, objective function, and expected output.	
2	20%	Explained the following clearly: (1) DP is a right choice to solve a given optimization problem (5%); (2) The key benefit of using DP is clearly stated. (5%); and (3) The problem solved is challenging enough with a sufficiently large problem size. (10%)	
3	20%	Provide clear algorithm descriptions of Top-down approach (5%); (2) Bottom-up approach (5%); (3) Recursive approach (5%); (4) Pseudo-code for all algorithms (3%); and (4) Example solutions (2%)	
4	20%	Clearly discussed: (1) Running times of three algorithms (two DPs and recursive algorithm) in Big-O, Ω, and Θ (15%); and (2) Performance comparison of the three algorithms (5%)	
5	15%	Demonstrated the following sensitivity analysis results: (1) Demonstrated experimental results of sensitivity analysis showing the performance under varying the values of the key design parameters (10%) – (one sensitivity analysis: 5%; 2 or more sensitivity analysis: 10%); and (2) Discussed the reasons of the observed results under the sensitivity analysis. (5%)	
6	10%	Provided proper answers for questions raised during the presentation.	
7	5%	Time management: Presented all prepared contents in a timely manner within 10 min. presentation time.	

Late penalty: 10% off per day (Any minute later after the deadline will be counted as one day late, e.g., 1 day late for the submission on 12:00am on the next day).

Project Examples

This is to help your brainstorming process. Hence, your project topic does not need to be restricted to the following topics as long as your project meets all the required conditions described in the above.

- Using DP and reinforcement learning to run a maze.
  https://medium.com/@zsalloum/basics-of-reinforcement-learning-the-easy-way-fb3a0a44f30e
  https://medium.com/@zsalloum/dynamic-programming-in-reinforcement-learning-the-easy-way-359c7791d0ac
- Sequence alignment for DNA in bioinformatics.
  https://en.wikipedia.org/wiki/Sequence_alignment
- Text segmentation (e.g. N-grams, Chinese) by using Viterbi.
  http://www.phontron.com/slides/nlp-programming-en-03-ws.pdf,
  https://github.com/willf/segment
- Optimizing IoT to Fog Latency in Resource Allocation.
  https://github.com/ismael-martinez/DynamicProgrammingProject2
- Mona Lisa TSP.
  http://www.math.uwaterloo.ca/tsp/data/ml/monalisa.html
- Implement a simple version of GNU diff.
  http://man7.org/linux/man-pages/man1/diff.1.html,
  https://wiki.c2.com/?DiffAlgorithm
- Comparing dynamic time warping (DTW) and RNN for voice classification.
  https://en.wikipedia.org/wiki/Dynamic_time_warping
- Pathfinding in a 3D space with 3D obstructions.
  https://en.wikipedia.org/wiki/A*_search_algorithm,
  https://www.youtube.com/watch?v=aKYlikFAV4k
- Seam carving for photos.
  https://en.wikipedia.org/wiki/Seam_carving
- Line wrap and word wrap for text editors.
  https://en.wikipedia.org/wiki/Line_wrap_and_word_wrap,
  http://defoe.sourceforge.net/folio/knuth-plass.html