**Design and Analysis of Algorithm (CSC4202)**

**Second Semester, 2022/2023**

**ASSIGNMENT 1**

**PROJECT**

**Learning outcomes:**

1. Search for and organize information in their form of an algorithm for a computer problem as the following: (LL)
   1. Analyze the problem to suggest a solution by choosing a suitable algorithm design technique.
   2. Develop skills to reason about and prove properties of algorithms such as their correctness and running time.

**Instructions:**

1. General instructions:
   1. Create an original scenario (you can be inspired by examples on the internet or used in the course, but the detailing of the story should be created fresh by your team of 4 people) that requires an optimal solution.
   2. Explain why finding an optimal solution for this scenario is important.
   3. Review the suitability of sorting, DAC, DP, greedy and graph algorithms as a solution paradigm for the chosen problem by stating their strengths and weaknesses.
   4. Design the algorithm to solve the problem and explain the idea of your algorithm paradigm by emphasising which part needs recurrence and the function for the optimization.
   5. Define the algorithm specification.
   6. Develop a program using any language of your convenience.
   7. Provide an analysis of the algorithm’s correctness as well as time complexity (best, average and worst time) by using asymptotic notation.
   8. Develop an online portfolio (using google sites or github or google colab or any suitable tool) with the following steps/content:
      1. Illustrate the problem.
      2. Explain your algorithm paradigm and show the pseudocode. You may provide your code in the portfolio if you wish.
      3. Demonstrate your program and describe the output.
      4. Describe the algorithm analysis.
   9. Deliver a presentation in week 14.
   10. Submit the following through Putrablast before the presentation
       1. Link to your online portfolio
       2. A zip file of your codes
       3. Filled project progress (refer APPENDIX)
2. Please make sure each member’s workload is fairly distributed and good project management is exercised. The weight of the project is 20%. Peer-based evaluation (5 marks) will be utilised besides the evaluations through filled progress monitoring (10 marks), online portfolio in github (50 marks) and presentation (15 marks).
3. Follow the following algorithm specification steps:  
   • Problem definition based on the chosen scenario (tell your story by choosing a geographical setting, type of disaster, damage impact, highlight the importance of AAD in the scenario and provide illustrations, state the goal and expected output to support the decision making)  
   • Development of a model for the chosen scenario (state the data type, state the objective function and constraints, provide examples and other requirements based on the scenario such as objective, space, time or value constraints)  
   • Specification of an Algorithm (state which topic and algorithm have you selected and why, include comparison of several other options and discuss the suitability of your proposed solution)  
   • Designing an Algorithm (provide a pseudocode and/or flowchart and use illustrations to help you)  
   • Checking the correctness of an Algorithm (asymptomatic, recurrence)  
   • Analysis of an Algorithm (growth of function for worst, best, average analysis)  
   • Implementation of an Algorithm (choose any language you prefer)  
   • Program testing (provide a demo based on your story of the chosen scenario)  
   • Documentation through online portfolio
4. Tips for completing the project
   1. Understanding of the AAD topics including examples of problems and list of algorithms that each topic covers is a MUST to ensure you can design the solution based on the instructions given.
   2. The “How Might We (HMW) technique” launches brainstorms by asking questions that seed your idea. Eg, HMW maximize the profit from building on this location by optimising the selection from a list of property options? From this question, your group may discuss potential situations and solutions. Then, discuss the possible algorithms to be used for this problem.
   3. You need to work collaboratively and be encouraged to use various materials around you as your reference (please make sure you include a good bibliography list in your documentation). During discussion, you may use ideas reviewing techniques such as identifying the “Pluses, Potential, Concerns, Options (PPCO)”.

Pluses: What are (at least) three things you like about the idea?

Potentials: What are (at least) three good things that might result if the idea were implemented?

Concerns: What are some concerns you have about the idea (phrased as a question starting with “How to…” or “How might…”)

Options or Overcome the concerns: What are some ideas you have for how to fix the concerns you just noted?

Or, you may use the Strength, Weakness, Opportunities, Threats (SWOT) technique.

* 1. Time management and each member’s dedication is the key to the group’s success.
  2. Use your creativity and critical thinking skills to design the algorithms and communicate it well in written and verbal format.

1. Some guidance for each AAD topic is as follows:
   1. Algorithm is a finite set of instructions that specify a sequence of operations to be carried out in order to solve a specific problem or class of problems called an algorithm. An algorithm is also an abstraction of a program to be executed on a physical machine (model computation).
   2. Generally, divide-and-conquer (DAC) algorithms have three parts −  
      • Divide the problem into a number of subproblems that are smaller instances of the same problem.  
      • Conquer the sub-problems by solving them recursively. If they are small enough, solve the sub-problems as base cases.  
      • Combine the solutions to the sub-problems into the solution for the original problem.  
      Example applications of DAC are merge sorting and matrix multiplication.
   3. Dynamic Programming (DP) algorithm is about always remembering answers to the sub-problems you’ve already solved. It is designed using the following four steps −  
      • Characterize the structure of an optimal solution.  
      • Recursively define the value of an optimal solution.  
      • Compute the value of an optimal solution, typically in a bottom-up fashion.  
      • Construct an optimal solution from the computed information.  
      DP is useful in optimisation and combinatorial problems such as Fibonacci numbers, recursion, shortest path and longest common subsequence.
   4. In Greedy Algorithm a set of resources are recursively divided based on the maximum, immediate availability of that resource at any given stage of execution. To solve a problem based on the greedy approach, there are two stages: Scanning the list of items and Optimization. Greedy algorithms have the following five components −  
      • A candidate set − A solution is created from this set.  
      • A selection function − Used to choose the best candidate to be added to the solution.  
      • A feasibility function − Used to determine whether a candidate can be used to contribute to the solution.  
      • An objective function − Used to assign a value to a solution or a partial solution.  
      • A solution function − Used to indicate whether a complete solution has been reached.  
      Example applications of greedy algorithms are activity selection, knapsack problem, minimum spanning tree, shortest path, Huffman code, and shortest path.
2. Useful references  
     
   <https://www.tutorialspoint.com/design_and_analysis_of_algorithms/design_and_analysis_of_algorithms_quick_guide.htm>  
     
     
   <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-046j-design-and-analysis-of-algorithms-spring-2015/>  
     
   <https://www.cet.edu.in/noticefiles/278_DAA%20Complete.pdf>

<https://algorithm-visualizer.org/>

APPENDIX

**Initial Project Plan (week 10, submission date: 26th May 2023)**

|  |  |
| --- | --- |
| Group Name |  |
| Members | |  |  |  | | --- | --- | --- | | Name | Email | Phone number | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |
| Problem scenario description | <give a rough idea on the problem> |
| Why it is important | <when and where the idea would be relevant, and what for?> |
| Problem specification |  |
| Potential solutions |  |
| Sketch (framework, flow, interface) |  |

**Project Proposal Refinement (week 11, submission date: 2nd June 2023)**

|  |  |
| --- | --- |
| Group Name |  |
| Members | |  |  | | --- | --- | | **Name** | **Role** | |  |  | |  |  | |  |  | |  |  | |
| Problem statement |  |
| Objectives |  |
| Expected output |  |
| Problem scenario description |  |
| Why it is important |  |
| Problem specification |  |
| Potential solutions |  |
| Sketch (framework, flow, interface) |  |
| Methodology | |  |  | | --- | --- | | Milestone | Time | | <eg: scenario refinement> | 26/5 (wk10) | | <eg: find example solutions and suitable algorithm. Discuss in group why that solution and the example problems relate to the problem in the project> | 2/6 (wk11) | | <eg: edit the coding of the chosen problem and complete the coding. Debug> | 9/6 (wk12) | | <eg: conduct analysis of correctness> | 16/6 (wk13) | | <eg: conduct analysis of time complexity> | 16/6 (wk13) | | <prepare online portfolio> | 23/6 (wk14) | | <prepare presentation> | 23/6 (wk14) | |

**Project Progress (Week 9 – Week 14)**

|  |  |
| --- | --- |
| **Milestone 1** |  |
| **Date (Wk)** |  |
| **Description/**  **sketch** |  |
|
| **Role** | |  |  |  |  | | --- | --- | --- | --- | | Member 1 | Member 2 | Member 3 | Member 4 | |  |  |  |  | |

|  |  |
| --- | --- |
| **Milestone 2** |  |
| **Date (Wk)** |  |
| **Description/**  **sketch** |  |
|
| **Role** | |  |  |  |  | | --- | --- | --- | --- | | Member 1 | Member 2 | Member 3 | Member 4 | |  |  |  |  | |