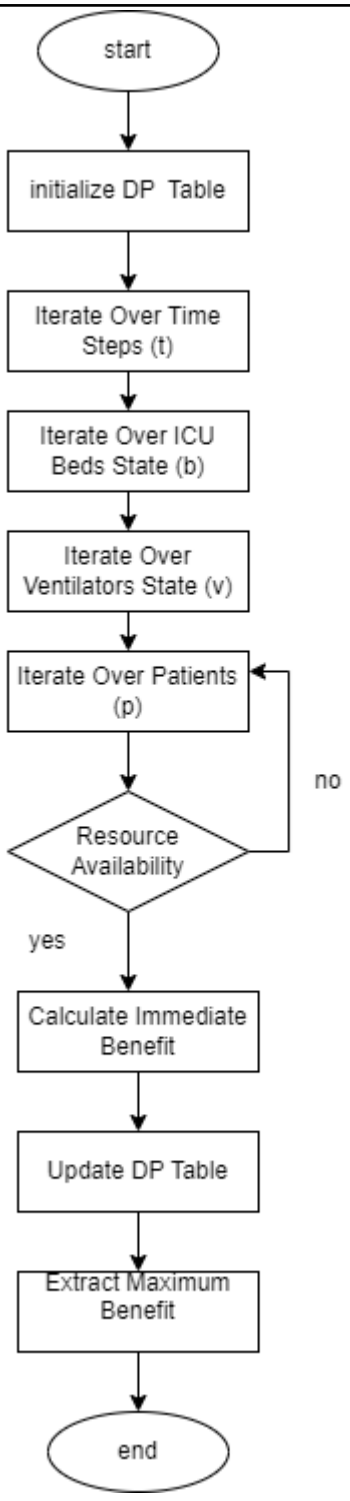
Design and Analysis of Algorithm (CSC4202)

Second Semester, 2023/2024

GROUP PROJECT
ASSIGNMENT 1

Initial Project Plan (week 10, submission date: 31 May 2024)

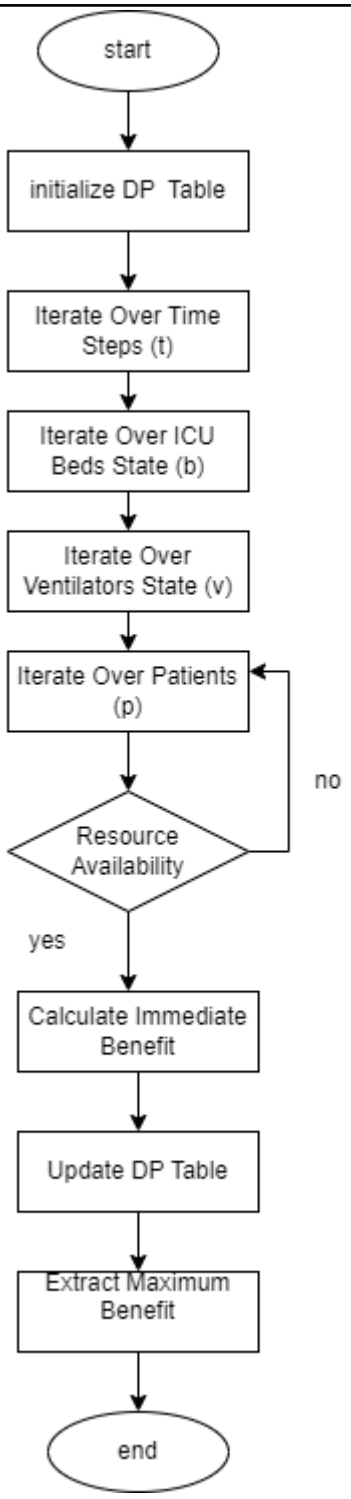
<table>
<tr><td>Group Name</td><td>AKI ALGORITHM</td></tr>
<tr><td>Members</td><td>
<table>
<tr><th>Name</th><th>Email</th><th>Phone number</th></tr>
<tr><td>NUR ADIBAH BINTI SAMSUL AZMAN</td><td>212225@student.upm.edu.y</td><td>01110586592</td></tr>
<tr><td>SITI KHADIJAH BINTI MOHD HAFIZ</td><td>211147@student.upm.edu.my</td><td>01111289422</td></tr>
<tr><td>IZZATUL SYAIRAH BINTI IBRAHIM</td><td>210196@student.upm.edu.my</td><td>01136811677</td></tr>
</table>
</td></tr>
<tr><td>Problem scenario description</td><td>During the height of a pandemic, hospitals are struggling with a surge of patients requiring critical care. Resources such as ICU beds, ventilators, and specialized medical staff are in short supply. Each patient’s condition varies, ranging from mild to severe, and their likelihood of survival with or without intensive treatment varies accordingly.</td></tr>
<tr><td>Why it is important</td><td>1. Maximizing Survival Rates
2. Ethical Considerations
3. Resource Utilization
4. Dynamic Response to Changing Conditions
5. Reducing Overload and Burnout
6. Public Trust and Compliance</td></tr>
<tr><td>Problem specification</td><td>1. Severity of Condition
2. Survival Probability
3. Resource Availability
4. Time Sensitivity</td></tr>
<tr><td>Potential solutions</td><td>Dynamic Programming (DP) is chosen for this problem due to its effectiveness in handling multi-faceted and time-dependent optimization problems. DP systematically breaks down the problem into simpler subproblems, solving each subproblem only once and storing its solution, which makes it efficient in terms of both time and space.</td></tr>
</table>

Sketch (framework, flow, interface)	

Project Proposal Refinement (week 11, submission date: 7 June 2023)

<table>
<tr><td>Group Name</td><td>AKI ALGORITHM</td></tr>
<tr><td>Members</td><td>
<table>
<tr><th>Name</th><th>Role</th></tr>
<tr><td>NUR ADIBAH BINTI SAMSUL AZMAN</td><td>Design algorithm</td></tr>
<tr><td>SITI KHADIJAH BINTI MOHD HAFIZ</td><td>Develop programming code</td></tr>
<tr><td>IZZATUL SYAIRAH BINTI IBRAHIM</td><td>Gather and compare suitable algorithm</td></tr>
</table>
</td></tr>
<tr><td>Problem statement</td><td>Hospitals face an overwhelming surge of patients needing critical care during a pandemic, with limited ICU beds, ventilators, and specialized staff. The challenge is to allocate these resources efficiently to maximize patient survival rates, considering varying conditions and survival probabilities, amidst dynamically changing resource availability and patient inflow.</td></tr>
<tr><td>Objectives</td><td>1. Maximize Survival Rates
2. Efficient Resource Use
3. Adapt to Changes</td></tr>
<tr><td>Expected output</td><td>Maximum Survival Benefit
The highest cumulative improvement in patient survival rates achievable by optimally allocating limited ICU beds and ventilators. This is calculated as the sum of the differences in survival probabilities with and without intensive care for all patients, considering the constraints of available resources and adjusting dynamically over time.</td></tr>
<tr><td>Problem scenario description</td><td>During the height of a pandemic, hospitals are struggling with a surge of patients requiring critical care. Resources such as ICU beds, ventilators, and specialized medical staff are in short supply. Each patient’s condition varies, ranging from mild to severe, and their likelihood of survival with or without intensive treatment varies accordingly.</td></tr>
<tr><td>Why it is important</td><td>1. Maximizing Survival Rates
2. Ethical Considerations
3. Resource Utilization
4. Dynamic Response to Changing Conditions
5. Reducing Overload and Burnout
6. Public Trust and Compliance</td></tr>
<tr><td>Problem specification</td><td>1. Severity of Condition
2. Survival Probability
3. Resource Availability
4. Time Sensitivity</td></tr>
<tr><td>Potential solutions</td><td>Dynamic Programming (DP) is chosen for this problem due to its effectiveness in handling multi-faceted and time-dependent optimization problems. DP systematically breaks down the problem into simpler subproblems, solving each subproblem only once and storing its solution, which makes it efficient in terms of both time and space.</td></tr>
</table>

Sketch (framework, flow, interface)



Methodology

Milestone	Time
Detailed Scenario Development and Initial Algorithm Review	wk10
Algorithm Design and Implementation	wk11

	Algorithm Specification and Pseudocode	wk12
	Algorithm Analysis and Example Scenario	wk13
	Presentation and Portfolio Preparation	wk14

Project Progress (Week 10)

Milestone 1	**Detailed Scenario Development and Initial Algorithm Review**
Date (week)	**01/06/2024 (Week 10)**
Description/ sketch	**1. Refine the detailed scenario for the algorithm's application.** **2. Discuss the importance of the scenario in solving the problem.**
Role	

Member 1	Member 2	Member 3
1. Research and refine scenario. 2. Analyze scenario importance.	1. Assist in refining the scenario. 2. Review and provide feedback.	1. Provide insights for scenario details. 2. Summarize the significance.

Project Progress (Week 11)

Milestone 2	**Algorithm Design and Implementation**
Date (Wk)	**03/06/2024 (Week 11)**
Description/ sketch	**1. Review and evaluate the suitability of existing algorithms for the given scenario.** **2. Design the dynamic programming (DP) algorithm specific to the scenario.** **3. Implement the designed DP algorithm in a suitable programming language.**
Role	

Member 1	Member 2	Member 3
1. Lead the design of the DP algorithm. 2. Implement core components. 3. Implement auxiliary functions.	1. Provide design input and critique. 2. Assist in evaluating algorithms. 3. Integrate and test the	1. Conduct algorithm suitability review. 2. Document the design process.

		complete algorithm.	3. Summarize findings in a report.

Project Progress (Week 12)

Milestone 3	**Algorithm Specification and Pseudocode**
Date (week)	**13/06/2024 (Week 12)**
Description/ sketch	**1. Continue implementation of the DP algorithm and refine based on initial tests.** **2. Develop detailed specifications and pseudocode for the algorithm.**
Role	

Member 1	Member 2	Member 3
1. Conduct further implementation and debugging. 2. Review and refine documentation	1. Assist in debugging and refining. 2. Write pseudocode based on implementation.	1. Test and validate intermediate results. 2. Draft detailed specifications.

Project Progress (Week 13)

Milestone 4	**Algorithm Analysis and Example Scenario**
Date (Wk)	**15/06/2024 (Week 13)**
Description/ sketch	**1. Analyze the correctness and complexity of the DP algorithm.** **2. Develop a detailed example scenario and illustrate the DP table step-by-step.**
Role	

Member 1	Member 2	Member 3
1. Perform correctness analysis. 2. Create example scenario.	1. Perform complexity analysis. 2. Illustrate DP table and process.	1. Review and compile analysis results. 2. Compile example into final document.

Project Progress (Week 14)

<table>
<tr><td>Milestone 5</td><td>Presentation and Portfolio Preparation</td></tr>
<tr><td>Date (Wk)</td><td>22/06/2024 (Week 14)</td></tr>
<tr><td>Description/ sketch</td><td>1. Prepare an online portfolio showcasing the project.
2. Prepare the final presentation.</td></tr>
<tr><td>Role</td><td>
<table>
<tr><td>Member 1</td><td>Member 2</td><td>Member 3</td></tr>
<tr><td>1. Design and structure the portfolio.
2. Practice and refine presentation.</td><td>1. Develop content for the portfolio.
2. Review and finalize the portfolio.</td><td>1. Create presentation slides.
2. Develop a presentation script.</td></tr>
</table>
</td></tr>
</table>