[Document title]



Data structures and Algorithm Designs

AS2\_CH1\_B1\_G6

05-Sep-2019

Venkataramanan Krishnan

Bala Vijay D

Bala Kavin Pon



02nd Assignment

bala vijay d

2018AC04514, Chennai Batch 1

Bala Kavin pon

2018AC04531, Chennai Batch 1

Venkataramanan Krishnan

2018AC04529, Chennai Batch 1

Created in collaboration of

Data structures

and

Algorithm

Designs

Table of Contents

[Problem Statement 4](#_Toc18427788)

[Requirement 4](#_Toc18427789)

[Deliverables 4](#_Toc18427790)

[Algorithm 5](#_Toc18427791)

[Greedy Approach 5](#_Toc18427792)

[Algorithm Design: 5](#_Toc18427793)

[Input: 5](#_Toc18427794)

[Output: 5](#_Toc18427795)

[Functional Elements: 5](#_Toc18427796)

# Problem Statement

There are **N** different models of mobiles manufactured at a mobile manufacturing unit.

Each mobile must go through 2 major phases:

1. Parts manufacturing
2. Assembling

Obviously, ‘parts manufacturing’ must happen before “assembling’. The time for ‘parts manufacturing’ and ‘assembling’ (***pmi*** and ***ai*** for ***ith*** mobile) for every mobile may be different. If we have only 1 unit for ‘parts manufacturing’ and 1 unit for ‘assembling’, how should we produce n mobiles in a suitable order such that the total production time is minimized?

# Requirement

1. Write a Greedy Algorithm to select the mobile parts manufacturing and assembling in such a way that total production time is minimized.
2. Analyse the time complexity of your algorithm.
3. Implement the above problem statement using Python.

# Deliverables

1. Word document designPS1\_<group id>.docx detailing your algorithm design and time complexity of the algorithm.
2. Zipped AS2\_PS1\_MM[Group id] package folder containing all the modules classes and functions for the employee node, binary tree and the main body of the program.
3. inputPS1.txt file used for testing
4. outputPS1.txt file generated while testing

# Algorithm

## Greedy Approach

A greedy algorithm approach is a simple, intuitive algorithm that is used in optimization problems. The algorithm makes the optimal choice at each step as it attempts to find the overall optimal way to solve the entire problem. The greedy method is applied to optimization problems, which involve searching through a set of configurations to find one that minimizes or maximizes an objective function defined on these configurations.

The given problem statement on scheduling the tasks in mobile manufacturing is analogous to the ***Task Scheduling*** problem in Greedy method paradigm. The conventional Task Scheduling problem is to schedule all the tasks in T on the fewest machines possible in a non-conflicting way. Whereas here, the given problem denotes that suppose a mobile has to be manufactured it needs to go through 2 stages of process, one is parts manufacturing(***pmi***) and second is assembling(***ami***) the manufactured parts. The assembly unit will have to wait for the parts manufacturing unit to finish and send it for assembling the parts. This will lead to an Idle time in the assembly unit when there are no mobile parts are finished with manufacturing or the manufacturing process take longer time. The objective of the algorithm is to find the optimal sequence of mobiles to be manufactured that will have minimum idle time of assembly unit.

This document describes the design of algorithm to solve problem of finding the optimal sequence of mobile to manufacture with minimal idle time of assembly units.

## Algorithm Design:

### Input:

The input is a list of tasks. The tasks will be in the structured form of triplets with a delimiter (**/**). For instance, **1/5/7**, each data in the triples will have to be mapped to ***Task ID, Part Manufacture Time*** and ***Assembling Time***.

### Output:

The expected output from the algorithm is to provide an optimal sequence of tasks that will take only minimal idle time in ***Assembly unit***, thus the production time will be minimized.

### Functional Elements:

The functional elements of the algorithm indicate the ***functionally significant logics*** that are need to complete the operations of the algorithm. There could be one or more such significant elements in an algorithm which plays a vital role in constructing the algorithm with minimal running time complexity.

There are two functionally significant elements required for constructing the algorithm,

1. Sorting the list of tasks based on parts manufacturing time and assemble time
2. Selection of tasks in sequence to have minimal idle time.