**Assignment\_PS10Q2 Design Document**

**Input :**

**Size of Input(N): No of products in Input file.**

For each product staging time and photo, shoot time is given.

**Output:**

The output must include the order in which products are handled, the total time to complete the photo shoot for all products, and idle time for photographer Xavier

**Algorithm:**

The studio has to complete both tasks in a defined order.

Staging -> Photoshoot i.e. if a product is not staged it cant be photographed.

In this situation, we have to choose the product which provides the most benefit (local optimum).

The products which can be setup quickly(lowest staging time) provide the most benefit as Xavier will not have to wait.

In situations where staging time is the same, the tiebreak can be resolved by choosing a product with the lowest photoshoot time.

After sorting the products on staging time and in case of tie the value of photoshoot time is used.

We create three arrays to track staging finish time, photo start time and idle time for all products

1. if the planned photo start time is greater the han staging finish time then xavier doesnt wait i.e. idle time for that product is 0
2. if the planned photo start time is less than the staging finish time then xavier hast to wait and the photo shoot is delayed
   * + - 1. Idle\_time[i]=Stage\_fin[i]-Photo\_start[i] for all i in product list
         2. Photo\_start[i]=Stage\_fin[i]

**Time complexity analysis:**

First operation is sorting to obtain sequence

* The sorting is done on staging time values [O(nlogn)]
* For ties in staging time values we go for values of photo time to break tie
* In the worst case all staging time values will be same and second sorying needs to be done in sequence [O(nlogn)]
* Total time nlogn+nlogn=2nlogn
* 2nlogn is still upper bounded by O(nlogn)

Second part involves calculating values for 3 arrays described in algorithm above

* For i in list of products
* Calculate values for arrays (constant time)
* Total time taken n\*(constant) which is upper bounded by O(n)

**Total time taken by algorithm:**

O(NlogN)+O(N) which is upper bounded by O(NlogN)

The boilerplate code to read file, prepare data and write to file takes constant time as both input and output files have 3 lines each. It doesn’t depend on the no of products.