**Query:**

1. Mark a selected worker as booked for a particular hour.
2. Delete an already marked worked as not booked for a particular hour.
3. At a particular day and time find the workers who are occupied. Same for the workers who are free.
4. For a particular day show status of a particular worker.

**Approach:**

Let us assume we have infinite memory:

*Terms:*

*n = total no. of days*

*m = total no. of workers*

Approach: For a particular day, we will maintain an (*24 x m)* size boolean matrix.

If a grid (i, j) is set (i.e have value 1) in the matrix then: *At ith hour jth worker is busy.*

If a grid (i, j) is unset (i.e have value 0) in the matrix then: *At ith hour jth worker is free.*

**Time complexity: (Theoretical)**

Now coming to the time consumed by each query in order:

1. Need to search the day, and then mark worker busy:

For searching day: O(log n)

For marking busy: O(1)

Total time: O(log n) \* O(1)

1. Need to search the day, and then mark worker free:

For searching day: O(log n)

For marking busy: O(1)

Total time: O(log n) \* O(1)

1. At a particular day and time, find who are occupied and who are free:

For searching the day: O(log n)

For getting the results: O(m)

Total time: O(log n) \* O(m)

1. For a particular day show stats of a particular worker:

For searching the day: O(log n)

For getting the results: O(24)

Total time: O(log n) \* O(24)

**Advantages:**

1. Since there is not any operation which is taking a quadratic time complexity (i.e. O(n2) ), system will have fast response times.
2. It can work for a small organisation and is simpler to code.

**Disadvantages:**

1. Total storage complexity becomes: n\*m\*24

This has several issues if the worker base becomes bigger, it will require a huge amount of space, irrespective of the action that is assigned to the user.

**Optimizations:**

1. We can reduce storage complexity, thus reducing our storage costs.

First, we can limit our ‘n’. We will only store data for a limited period, like 180 days etc. Rest of data can be discarded or dumped in a separate location.

1. We will not create an static array of m\*24, this is a huge overhead, we will use hashmap instead for dynamic allocation of that ‘m’. With this huge computational and storage load will be reduced, as only those workers which have atleast one meeting will be present there.