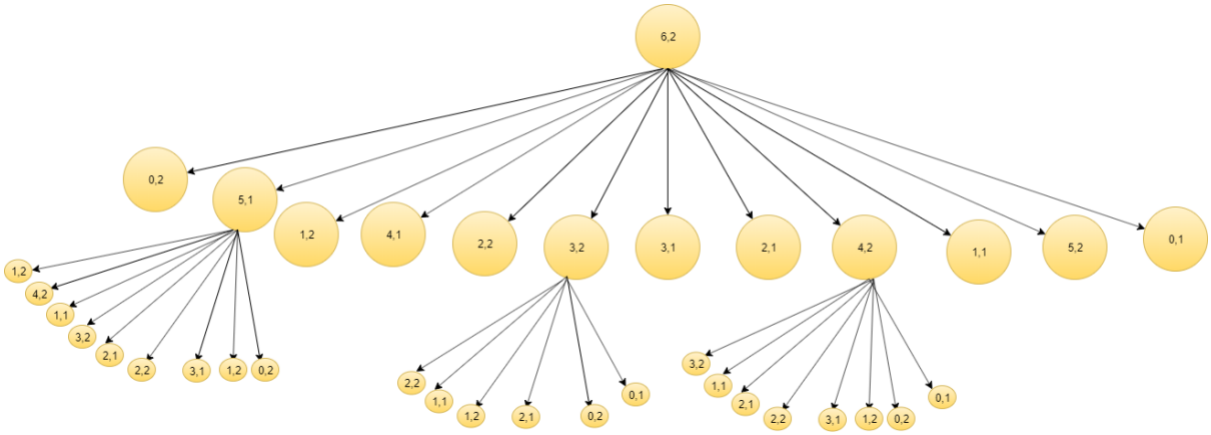
Running Trials and Tribulations

A training program has been developed which suggests the athletes to train for their race on a certain pace in order to prevent injuries during the actual race. The athlete must input the number of days and will provide range of distinct running speeds from which, the program must pick the fastest speed that an athlete can train for while eliminating the possibility of any injury during the actual race. The program will list the speed range provided by athlete as 1^{st}, 2^{nd}, 3^{rd}, ……, n^{th} speed. Based on the circumstances, if an athlete enters only a single day for training then the program must recommend the 1^{st} speed and it should keep it incrementing by 1 till the point risk of injury is not persistent. The moment, risk of injury is sensed, the program must stop incrementing and provide a final speed which must not be exceeded by any chance. If there are more than one day available for training, an athlete must ideally want to train at the same speed in order to do the condition of body. The program suggests the athlete that training conditions must remain same, the only component that should be varying is the speed of running. Also, if an athlete gets injured on a particular speed, they must spend rest of the day resting and must only continue training the next day. Moreover, athlete must not exceed the speed at which the injury had occurred. Although, if athlete tries out a new speed, then training must be continued at the same speed for rest of the day in order to condition the muscles for that speed, so that the speed can be easily maintained on the race day itself without any possibility of injury.

a) **Recurrence –** Recurrence can be termed as a repeated occurrence of the same instance which has happened before. A function calls itself again and again until the base condition is satisfied.
b) **Recurrence Tree –** It is a tree which is used for solving recurrence relations. In this tree, each node represents cost of recursive sub-problem. In simple words, it is a pictorial representation of iterations in the form of a tree, where nodes are expanded at every level.
c) **Drawing Recurrence Tree**



d) How many distinct subproblems do you end up with given 6 speeds and 2 weeks?

12

e) How many distinct subproblems for N speeds and M days?

M*N subproblems are there for N speeds and M days.

Holiday Special – Putting Shifts Together

This algorithm is for solving the complexity of the community for the recipes. There will be a volunteer who will be cooking for m number of people and for every distinct recipe, there will be distinct subset of steps that needs to be performed in specific orders. Longer shifts must be implemented for cooks in order to run the work in a seamless manner. In this, a system is designed which will allot the steps to the staff for a selected recipe and will automatically make the schedule which might need least number of people switching in-between.

a) Substructure – A problem is said to be a substructure if its solution can be obtained from the solution of its sub-problem. It is a property which is primarily used for determining the usefulness of greedy algorithm and dynamic programming. For substructure of given problem, the program takes cooks with highest number of consecutive steps.

b) Greedy Algorithm – Greedy algorithm is basically a strategy which makes optimal choice at every small stage in order to achieve a global optimum solution for a problem. This algorithm picks the best solution regardless of the consequences which are associated with it.

c) Runtime Complexity and Scheduling Algorithm – The runtime complexity of an algorithm is the amount of time taken to generate appropriate output based on the input given.

Whereas, scheduling algorithm decides the priority on which the processes will be executed. Scheduling is basically of two types i.e. pre-emptive scheduling and non-pre-emptive scheduling. There are multiple scheduling algorithms namely First Come First Serve (FCFS),

Shortest Job First (SJF) priority scheduling, round robin scheduling, multilevel queue scheduling and multi-level feedback queue scheduling.

(d) The runtime complexity of greedy algorithm

O(nm)+O(nm) where n= number of cook and m = steps

League of Patience

In this, a scenario is given in which a system must be developed which will tell the exact route which must be taken for the quests. It starts from the starting point of the client and the quest to which the client wants you to get them. In this system, there is no bar for the number of quests that needs to be completed. Although, some of the quests are shorter while some quests are considerably longer. The system must depict the fastest overall time which will be taken for starting location to end point.

(a) An algorithm solution to this problem

To sort this problem out, the program uses Dijkstra's algorithm.

(b) The complexity of your proposed solution in (a)

O(V^2) + O(V) where V= node

(c) Algorithm is this genericShortest method implementing

Dijkstra's Algorithm is used in genericShortest method.

(d) To complete the quest the weight of edges is being updated.

(e) The algorithm used in the implementation takes O(V^2) to run.