Binary Search Advanced Problem Solving

By- Priyansh Agarwal

Requirement for using Binary Search

Monotonicity

- f(x) > f(y) iff x > y (increasing monotonic)
- f(x) < f(y) iff x > y (decreasing monotonic

Binary Search on Answer (Important + Non

- Consider a predicate P defined over some ordered set S (the search space). The search space consists of candidate answers to the problem. In our case, a predicate is a function which returns TRUE or FALSE. We use the predicate to verify if a candidate answer is legal or not.

- Example: We have the set of numbers {1, 2, 3, 4, 5}. Our predicate function could be following:
  Return TRUE if the number is less than 3 and FALSE otherwise
  Now, if we pass 2 to this function, it will return TRUE right?

Condition on Predicate Required for doing Binary Search on Answer

- P(X) is TRUE => P(Y) is TRUE for all Y > X
  && P(X) is FALSE => P(Y) is FALSE for all Y < X

F	F	F	F	T	T	T	T	T	T	T

Or

- P(X) is TRUE => P(Y) is TRUE for all Y < X
  && P(X) if FALSE => P(Y) is FALSE for all Y > X

T	T	T	T	T	F	F	F	F	F	F

FairWorkload Problem

- Given an array of workloads, split it among ‘k’ workers, such that the maximum work that any worker has to do is minimised (can’t change order of workloads).
- Eg. 10 20 30 40 50 60 70 80 90
  - Solution : 10 20 30 40 50 | 60 70 | 80 90

    First worker - 150,

    Second worker - 130,

    Third worker - 170
- Is it possible to partition workload in a way that the highest workload of any worker is less than 170? Hence, answer is 170.

FairWorkload Problem

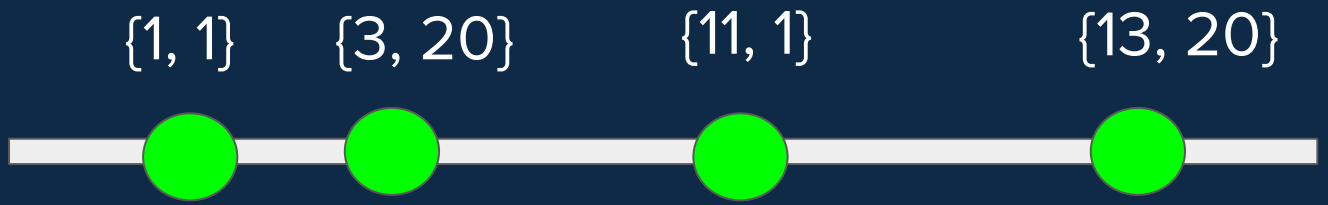
Given an array of workloads, split it among ‘k’ workers, such that the maximum work that any worker has to do is minimised (can’t change order of workloads).

Think about it: For any given ‘workload’ can we check if this is valid for given k or not?

Get Together Problem

There are n people standing in a line. Each of them has a X_i (Current Position) and V_i (Maximum Speed allowed). All of them want to gather at a single point. Find the minimum time required for them to do so.

Eg.



Ans: It is optimal that the first person starts moving to the left and the 3rd person starts moving to the right. They will meet at 6 in 5 seconds and other 2 people would also be able to reach 6 in less than 5 seconds (obvious looking at their speed). So, the answer is 5 seconds

Interactive Problems: Introduction

Bonus Problems

Binary Search Twisted:

1) Binary Search on a Rotated Sorted-Array (Distinct Elements)
2) Find Peak Element in a Bitonic Array (Distinct Elements)

Binary Search on Answer:

1) Aggressive Cows Problem
2) Guess the Greatest Element - Interactive Problem