Interval Partitioning

Refer: Chap 4 Tardos

10-02-2022

Interval Scheduling - Recall

- We are given a set of job/interval requests, $S = a_1, a_2, . . . , a_n$ that need to use some resource.
- Each request a_i has a start time s_i & finish time f_i, such that $0 \leq s_i < f_i < \infty$.
- We need to allocate the resource in a compatible manner, such that the number of request scheduled is maximized.
- The resource can be used by one and only one request at any given time.
- Eg: Entertaining student queries in the admin office of SGTB Khalsa college

Interval Partitioning

Interval Partitioning - Introduction

Given a set of intervals formatted as (s_i, f_i) and a **collection** of similar resources.

Goal: Schedule **all** intervals **minimizing** the number of

Resources.

The problem boils down to finding the minimum number of resource instance needed to schedule all jobs.

Eg: Classes in SGTB Khalsa college at 8:30 AM.

Why partitioning?

Partition the jobs/interval into groups based on the resource they acquire.

Example:

How many minimum resources do you need to schedule ALL jobs?

Why partitioning?

Partition the jobs/interval into groups based on the resource they acquire.

Example:

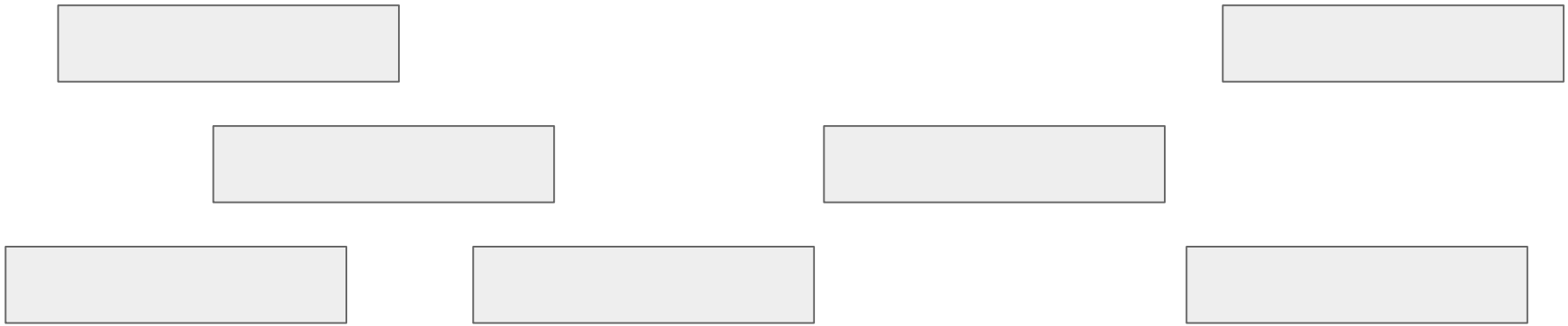
How many minimum resources do you need to schedule ALL jobs?

Answer: 2

Partition intervals into 2 groups based on resource instance.

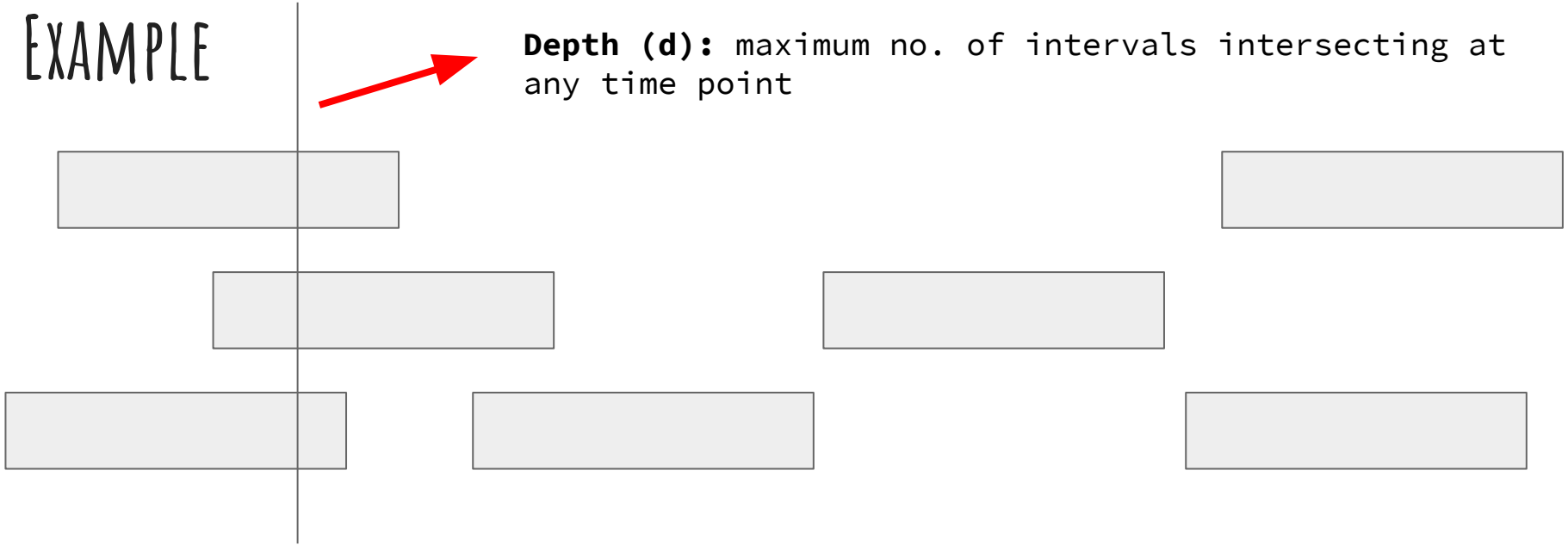
Job 1 -> instance 1
Job 2 -> instance 2
Job 3 -> instance 1 or 2

Example



Can you tell how many minimum resources are needed to schedule all intervals in this case?

Example



Depth (d): maximum no. of intervals intersecting at any time point

Can you tell how many minimum resources are needed to schedule all intervals in this case? **3 = d**

Clearly, at least d resources are required to schedule them i.e. d is a lower bound for OPT.

Interval colouring

The same problem is also called interval colouring.

How many colors will you need if a job can use one color and no other job which is non-compatible can use that same color?

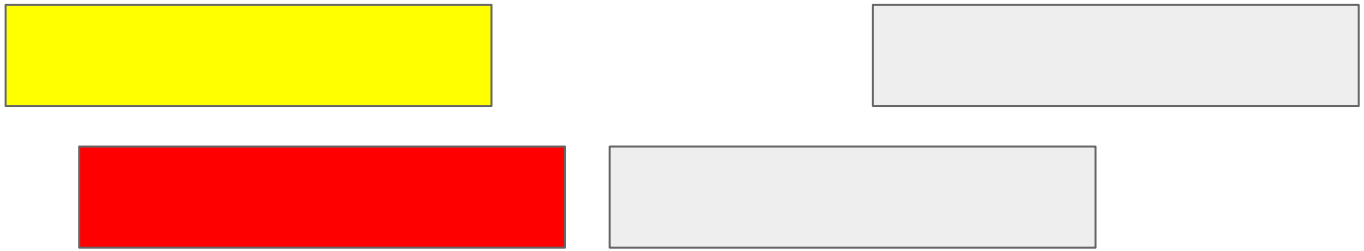
Simplest: use 4 colours.

Interval colouring

How many colors will you need if a job can use one color and no other job which is non-compatible can use that same color?

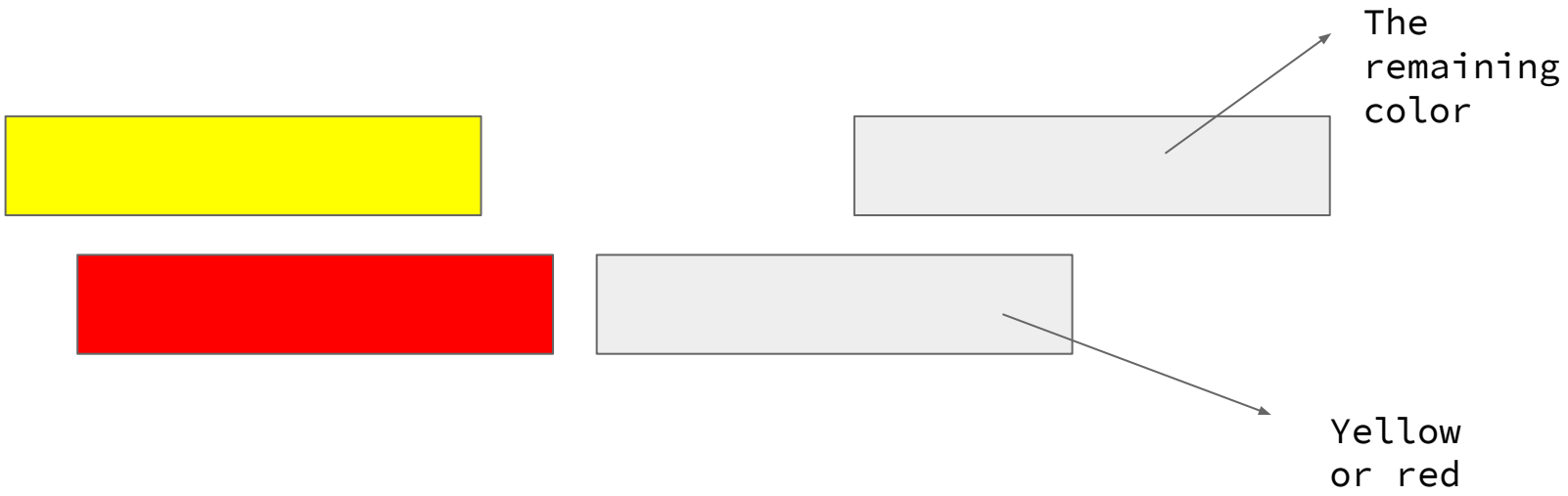
Interval colouring

How many colors will you need if a job can use one color and no other job which is non-compatible can use that same color?



Interval colouring

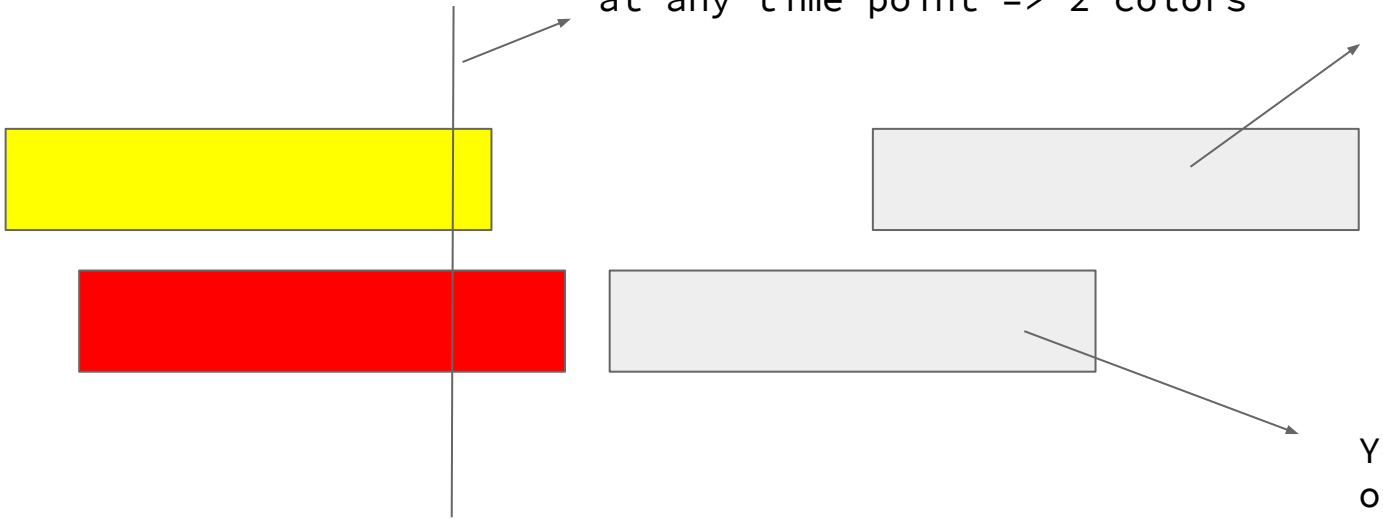
How many colors will you need if a job can use one color and no other job which is non-compatible can use that same color?



Interval colouring

How many colors will you need if a job can use one color and no other job which is non-compatible can use that same color?

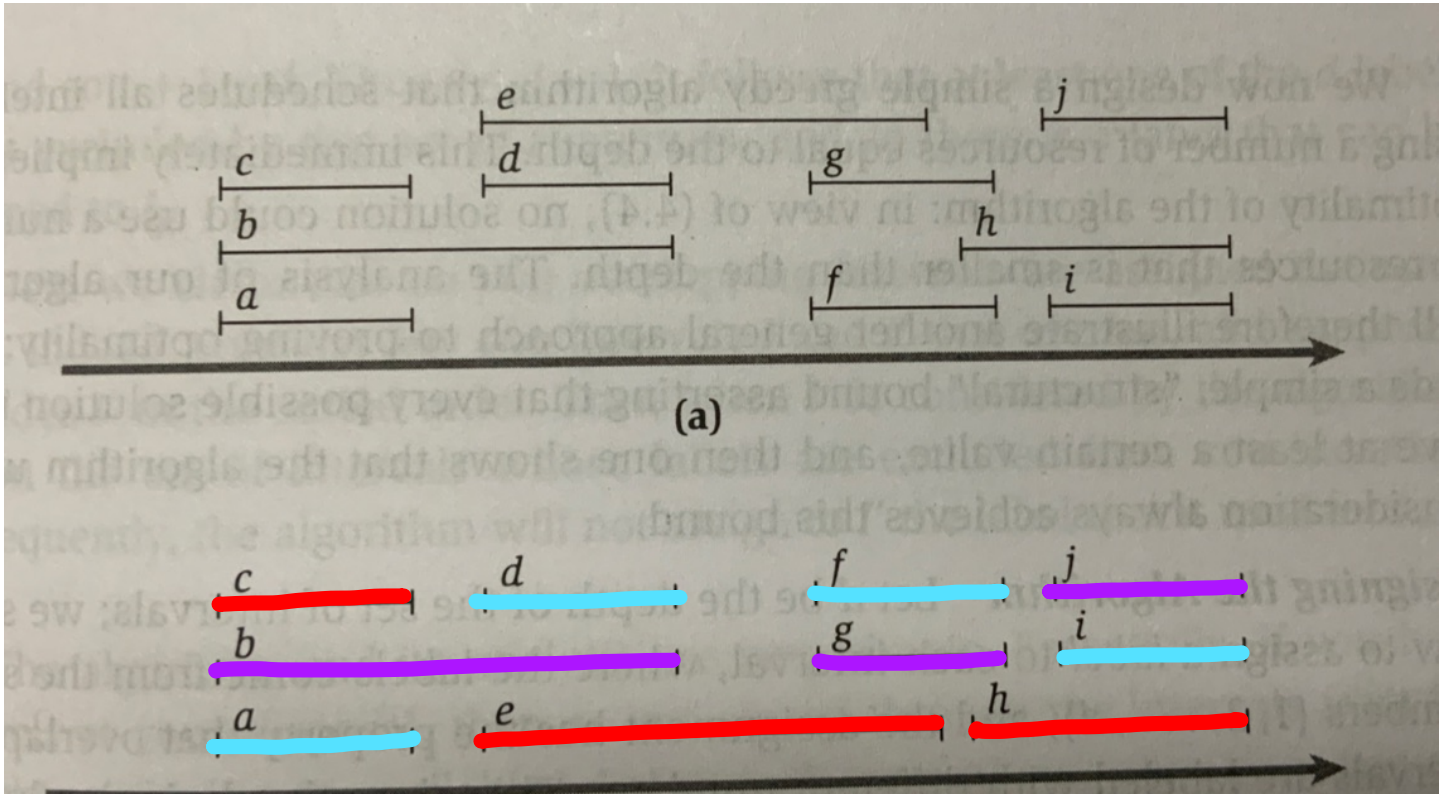
Depth (d): maximum no. of intervals intersecting at any time point => 2 colors



Algorithm - let us bend backwards

```
interval_partition(P,L)
// ‘P’ is the set of intervals, L={1, 2, . . . , d} are the colours
{
    Sort the intervals in increasing order of starting time;
    Pick any interval ‘i’ in P;
    Remove the colour of the conflicting intervals from consideration;
    Colour ‘i’ with any available colour;
}
```

Example



Proof of Correctness

Claim: (a) Every interval will be assigned a label/color, and (b) no two overlapping intervals will receive the same label/color.

(a) Let us consider an interval I_j and suppose there are t interval earlier than I_j in the sorted sequence that overlap it. They together form t+1 intervals that overlap in time. So, $t+1 \leq d$. Thus, $t \leq d-1$

This means that one of the d colors/labels is not excluded by this set of t intervals, and so there is a label that can be assigned to I_j.

Proof of Correctness

Claim: (a) Every interval will be assigned a label/color, and (b) no two overlapping intervals will receive the same label/color.

(b) Consider I and I' that overlap. And let us assume I comes before I' in the sorted order. When considering I', I is in the set of intervals whose labels are excluded from consideration. Therefore, the label for I will not be used for I'.

↳ By design of our algorithm

Summary

The greedy algorithm is using d labels, and upon sweeping through the intervals from L-to-r, assigning a label to each interval you encounter, we are using d resources. Here, d is the depth of the set of intervals. Also, the no. of optimal resources required.