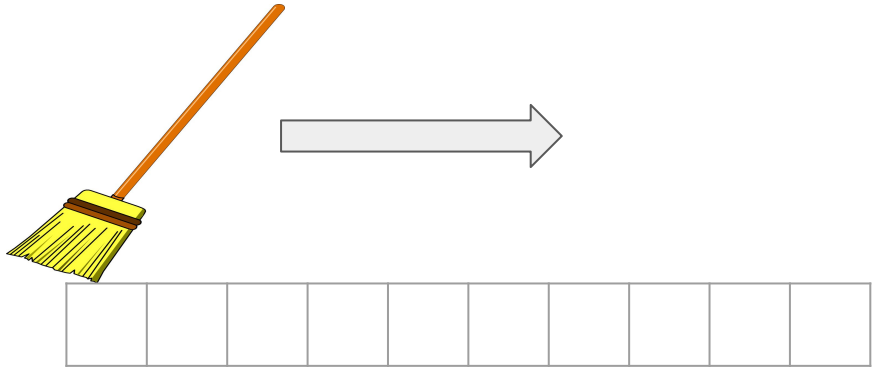
Greedy Sweeps

Sweep

Goal: start at one end of the data move across and build up the answer as we go.

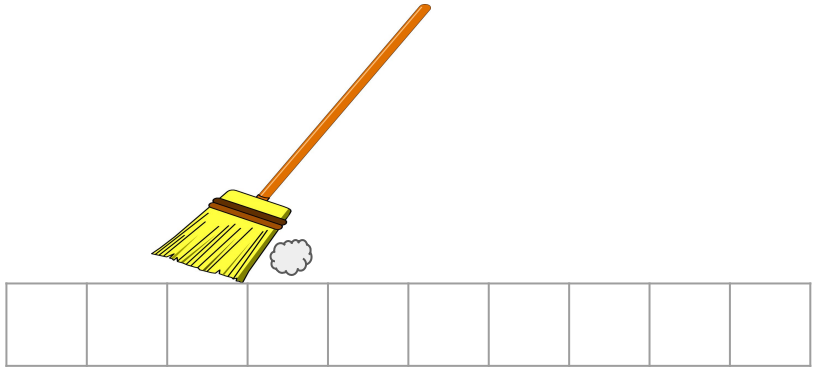
Sweep

Goal: start at one end of the data move across and build up the answer as we go.



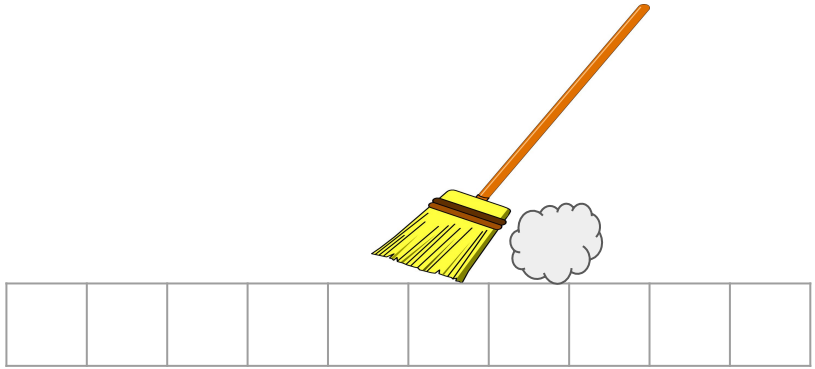
Sweep

Goal: start at one end of the data move across and build up the answer as we go.



Sweep

Goal: start at one end of the data move across and build up the answer as we go.



Sweep

Goal: start at one end of the data move across and build up the answer as we go.

Sweep

Goal: start at one end of the data move across and build up the answer as we go.

We need to ensure that the data is in the correct order.

Sweep

Goal: start at one end of the data move across and build up the answer as we go.

We need to ensure that the data is in the correct order.

The data needs to “sorted”

Example Problem

I need to tell all my students about some deadline

Example Problem

I need to tell all my students about some deadline (General Election Registration)

Example Problem

I need to tell all my students about some deadline (General Election Registration)

B/C of remote learning not all the students are in the zoom call at the same time T_T

Example Problem

I need to tell all my students about some deadline (General Election Registration)

B/C of remote learning not all the students are in the zoom call at the same time T_T

I know when each student will enter and leave the meeting (assume no student will re-enter)

Example Problem

I need to tell all my students about some deadline (General Election Registration)

B/C of remote learning not all the students are in the zoom call at the same time T_T

I know when each student will enter and leave the meeting (assume no student will re-enter)

What is the minimum number of times I need to mention the deadline to ensure all students are aware?

Observations/Ideas

Working with an array of students could be challenging

Observations/Ideas

Working with an array of students could be challenging

Turn students into intervals of when they will be available.

Observations/Ideas

Working with an array of students could be challenging

Turn students into intervals of when they will be available.

Note: each interval is a different student.

Observations/Ideas

Working with an array of students could be challenging

Turn students into intervals of when they will be available.

Note: each interval is a different student.

At a point in time we can give students the information.

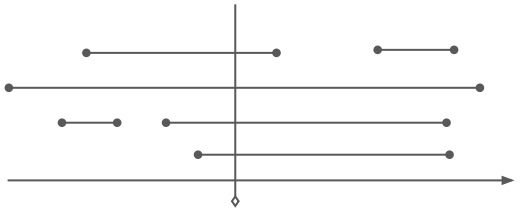
Observations/Ideas

Working with an array of students could be challenging

Turn students into intervals of when they will be available.

Note: each interval is a different student.

At a point in time we can give students the information.



Observations/Ideas

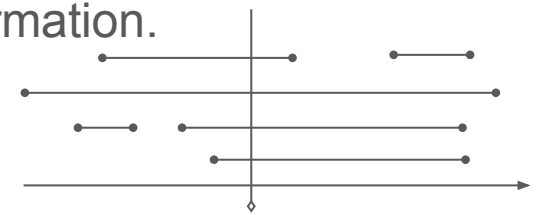
Working with an array of students could be challenging

Turn students into intervals of when they will be available.

Note: each interval is a different student.

At a point in time we can give students the information.

Those students will no longer need the information.



Observations/Ideas

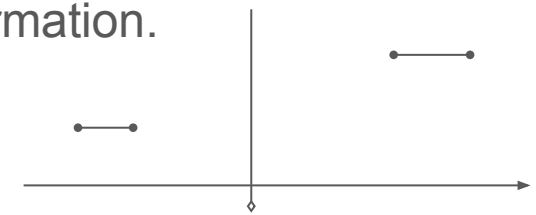
Working with an array of students could be challenging

Turn students into intervals of when they will be available.

Note: each interval is a different student.

At a point in time we can give students the information.

Those students will no longer need the information.



Observations/Ideas

Working with an array of students could be challenging

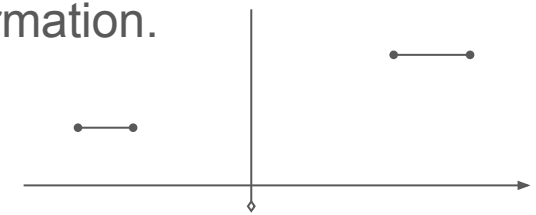
Turn students into intervals of when they will be available.

Note: each interval is a different student.

At a point in time we can give students the information.

Those students will no longer need the information.

Repeat until no student uninformed.



Observations/Ideas

Working with an array of students could be challenging

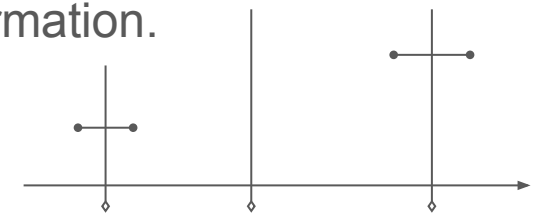
Turn students into intervals of when they will be available.

Note: each interval is a different student.

At a point in time we can give students the information.

Those students will no longer need the information.

Repeat until no student uninformed.



Optimizing Answers

How do we choose the best time to place these information points?

Optimizing Answers

How do we choose the best time to place these information points?

Largest overlap?

Optimizing Answers

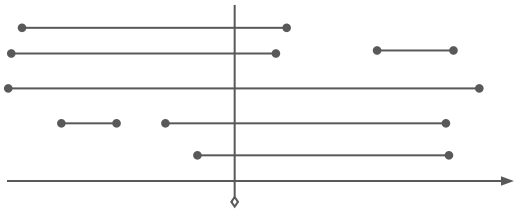
How do we choose the best time to place these information points?

Largest overlap? No.

Optimizing Answers

How do we choose the best time to place these information points?

Largest overlap? No.

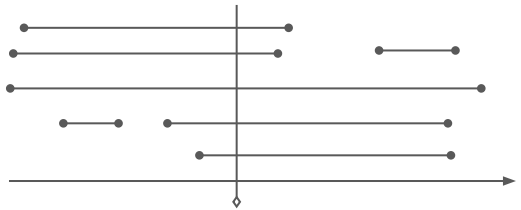


Optimizing Answers

How do we choose the best time to place these information points?

Largest overlap? No.

Consider the following,



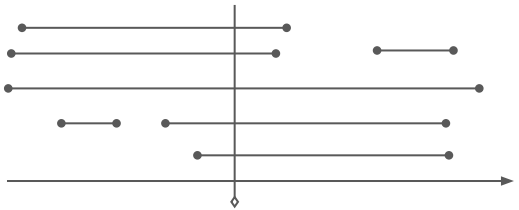
Optimizing Answers

How do we choose the best time to place these information points?

Largest overlap? No.

Consider the following,

> When a student leaves we must have informed them beforehand.



Optimizing Answers

How do we choose the best time to place these information points?

Largest overlap? No.

Consider the following,

> When a student leaves we must have informed them beforehand.

Be lazy

Optimizing Answers

How do we choose the best time to place these information points?

Largest overlap? No.

Consider the following,

When a student leaves we must have informed them beforehand.

Be lazy

Wait until an uninformed student is just about to leave, and then hit everyone with some knowledge

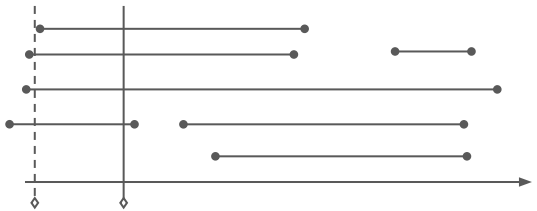
Proving This is Correct

We need to perform an exchange argument.

Proving This is Correct

We need to perform an exchange argument.

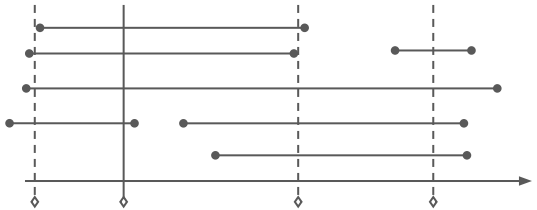
- “FTSOC Suppose an ideal solution performed a knowledge sesh earlier”



Proving This is Correct

We need to perform an exchange argument.

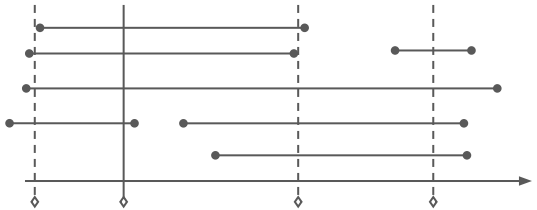
- “FTSOC Suppose an ideal solution performed a knowledge sesh earlier”
- We would show that we have an ideal solution could exists by moving forward the time by patchworking with the ideal solution.



Proving This is Correct

We need to perform an exchange argument.

- “FTSOC Suppose an ideal solution performed a knowledge sesh earlier”
- We would show that we have an ideal solution could exists by moving forward the time by patchworking with the ideal solution.
- Thus the ideal solution should do a later knowledge session.



Another Problem: Food Display Arrangement

Your friend works at a local grocery store and has the task of arranging produce. At the end of the day your friend needs to fix the order of the food. All the customers move the produce around (without buying it those jerks), and some lazy coworkers dumped all the new food in one spot.

Another Problem: Food Display Arrangement

Your friend works at a local grocery store and has the task of arranging produce. At the end of the day your friend needs to fix the order of the food. All the customers move the produce around (without buying it those jerks), and some lazy coworkers dumped all the new food in one spot.

- Each food has some numeric type (e.g. 7, 8, 3).

7	7	8	7	3	8

Another Problem: Food Display Arrangement

Your friend works at a local grocery store and has the task of arranging produce. At the end of the day your friend needs to fix the order of the food. All the customers move the produce around (without buying it those jerks), and some lazy coworkers dumped all the new food in one spot.

- Each food has some numeric type (e.g. 7, 8, 3).
- Your friend needs to have all the food of the same type “grouped” together in the resulting display.

7	7		7	3	8	8

Another Problem: Food Display Arrangement

Your friend works at a local grocery store and has the task of arranging produce. At the end of the day your friend needs to fix the order of the food. All the customers move the produce around (without buying it those jerks), and some lazy coworkers dumped all the new food in one spot.

- Each food has some numeric type (e.g. 7, 8, 3).
- Your friend needs to have all the food of the same type “grouped” together in the resulting display.
- In one pass your friend can walk along the display and pick up all the food of a certain type, dumping it at the end.

7	7		7	3	8	8

Another Problem: Food Display Arrangement (cont.)

- Each food has some numeric type (e.g. 7, 8, 3).
- Your friend needs to have all the food of the same type “grouped” together in the resulting display.
- In one pass your friend can walk along the display and pick up all the food of a certain type, dumping it at the end.

Another Problem: Food Display Arrangement (cont.)

- Each food has some numeric type (e.g. 7, 8, 3).
- Your friend needs to have all the food of the same type “grouped” together in the resulting display.
- In one pass your friend can walk along the display and pick up all the food of a certain type, dumping it at the end.

What is the least number of passes required by your friend?

FDA Strategy

How could we solve this greedily?

FDA Strategy

How could we solve this greedily?

Turn each food type into an interval!

FDA Strategy

How could we solve this greedily?

Turn each food type into an interval!

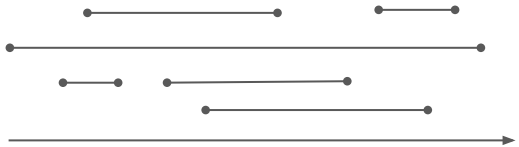
- Start is the first location
- End is the last location

FDA Strategy

How could we solve this greedily?

Turn each food type into an interval!

- Start is the first location
- End is the last location



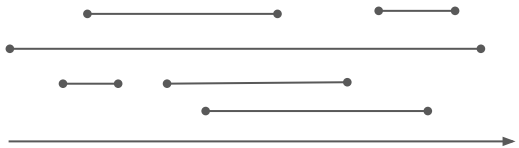
FDA Strategy

How could we solve this greedily?

Turn each food type into an interval!

- Start is the first location
- End is the last location

Food that will be left alone cannot overlap with other left alone food.



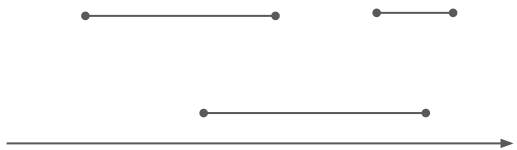
FDA Strategy

How could we solve this greedily?

Turn each food type into an interval!

- Start is the first location
- End is the last location

Food that will be left alone cannot overlap with other left alone food.



FDA Strategy

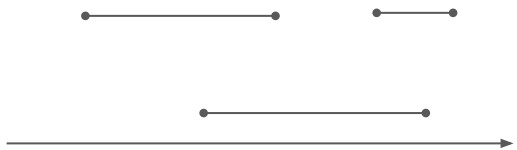
How could we solve this greedily?

Turn each food type into an interval!

- Start is the first location
- End is the last location

Food that will be left alone cannot overlap with other left alone food.

INVALID =>



FDA Strategy

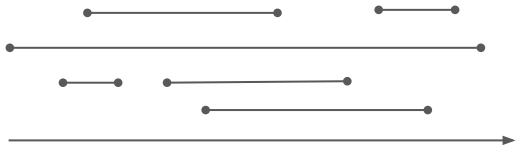
How could we solve this greedily?

Turn each food type into an interval!

- Start is the first location
- End is the last location

Food that will be left alone cannot overlap with other left alone food.

The more intervals left,
the less passes needed!



FDA Strategy

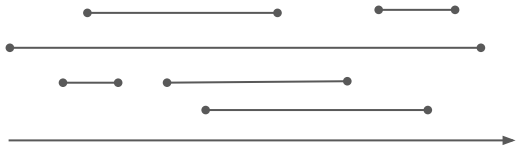
How could we solve this greedily?

Turn each food type into an interval!

- Start is the first location
- End is the last location

Food that will be left alone cannot overlap with other left alone food.

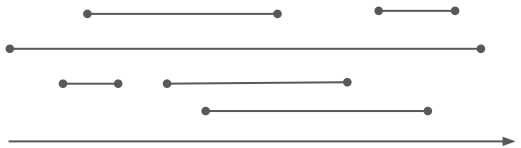
The more intervals left,
the less passes needed!



Find maximum non-overlapping subset.

Non-Overlapping Subset Strategy

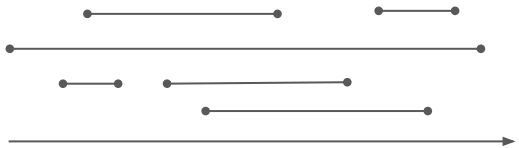
How do we do this?



Non-Overlapping Subset Strategy

How do we do this?

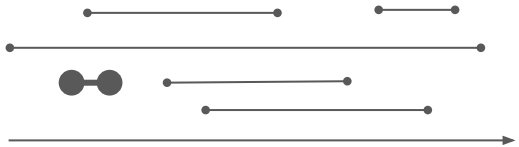
Taking the interval that ends the earliest is a good idea.



Non-Overlapping Subset Strategy

How do we do this?

Taking the interval that ends the earliest is a good idea.



Non-Overlapping Subset Strategy

How do we do this?

Taking the interval that ends the earliest is a good idea.

Non-Overlapping Subset Strategy

How do we do this?

Taking the interval that ends the earliest is a good idea.

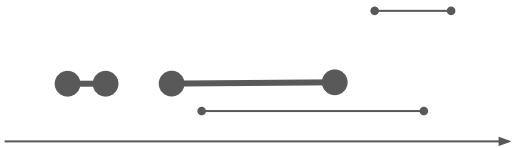
Repeat…

Non-Overlapping Subset Strategy

How do we do this?

Taking the interval that ends the earliest is a good idea.

Repeat…

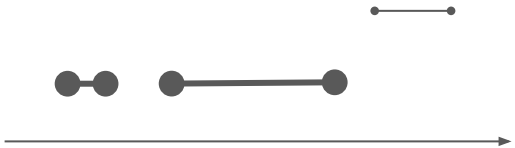


Non-Overlapping Subset Strategy

How do we do this?

Taking the interval that ends the earliest is a good idea.

Repeat…

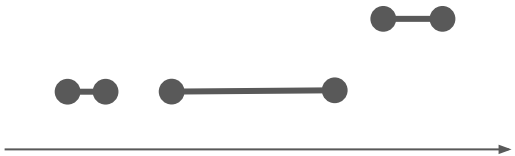


Non-Overlapping Subset Strategy

How do we do this?

Taking the interval that ends the earliest is a good idea.

Repeat…



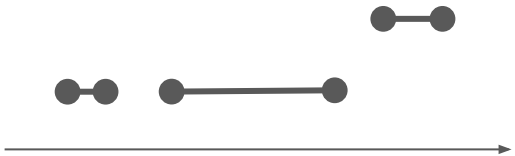
Non-Overlapping Subset Strategy

How do we do this?

Taking the interval that ends the earliest is a good idea.

Repeat…

3 food types can be left alone.



Non-Overlapping Subset Strategy

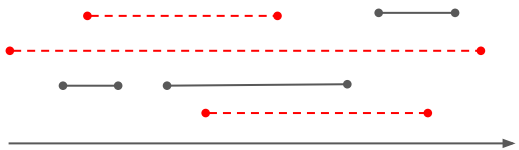
How do we do this?

Taking the interval that ends the earliest is a good idea.

Repeat…

3 food types can be left alone.

3 types had to move



High-Level

```
Convert to intervals
Sort by End Points (smallest first)
Make the Last End Point Used (LEPU) be -infinity
Let movedIntervals be 0
Loop through sorted intervals
   If the currentInterval.start is after the LEPU
      Update LEPU to currentInterval.end
   Else
      Increment movedIntervals by 1
   End If
End Loop
Return movedIntervals
```