

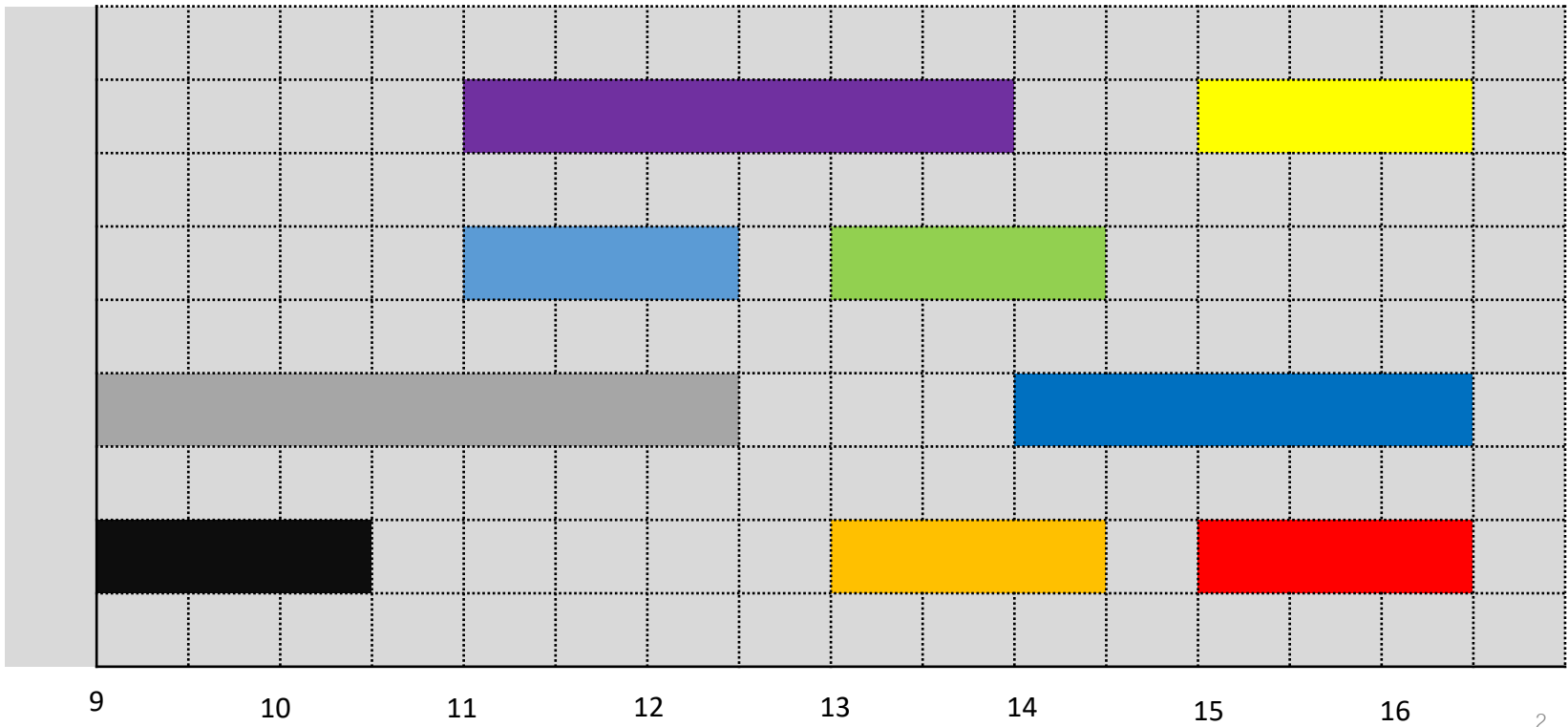
# Interval partitioning

# Interval partitioning

กำหนดให้ มีรายวิชาหลายๆ รายวิชาที่มีการบอกเวลาเริ่มเรียนและเวลาเลิกเรียน

Goal: หาจำนวนห้องที่น้อยที่สุดที่ต้องเปิดเพื่อทำตารางเวลาการใช้ห้องของรายวิชาโดยที่ไม่มีรายวิชาสองรายวิชาใดๆ ที่เรียนเวลาเดียวกันในห้องเดียวกัน

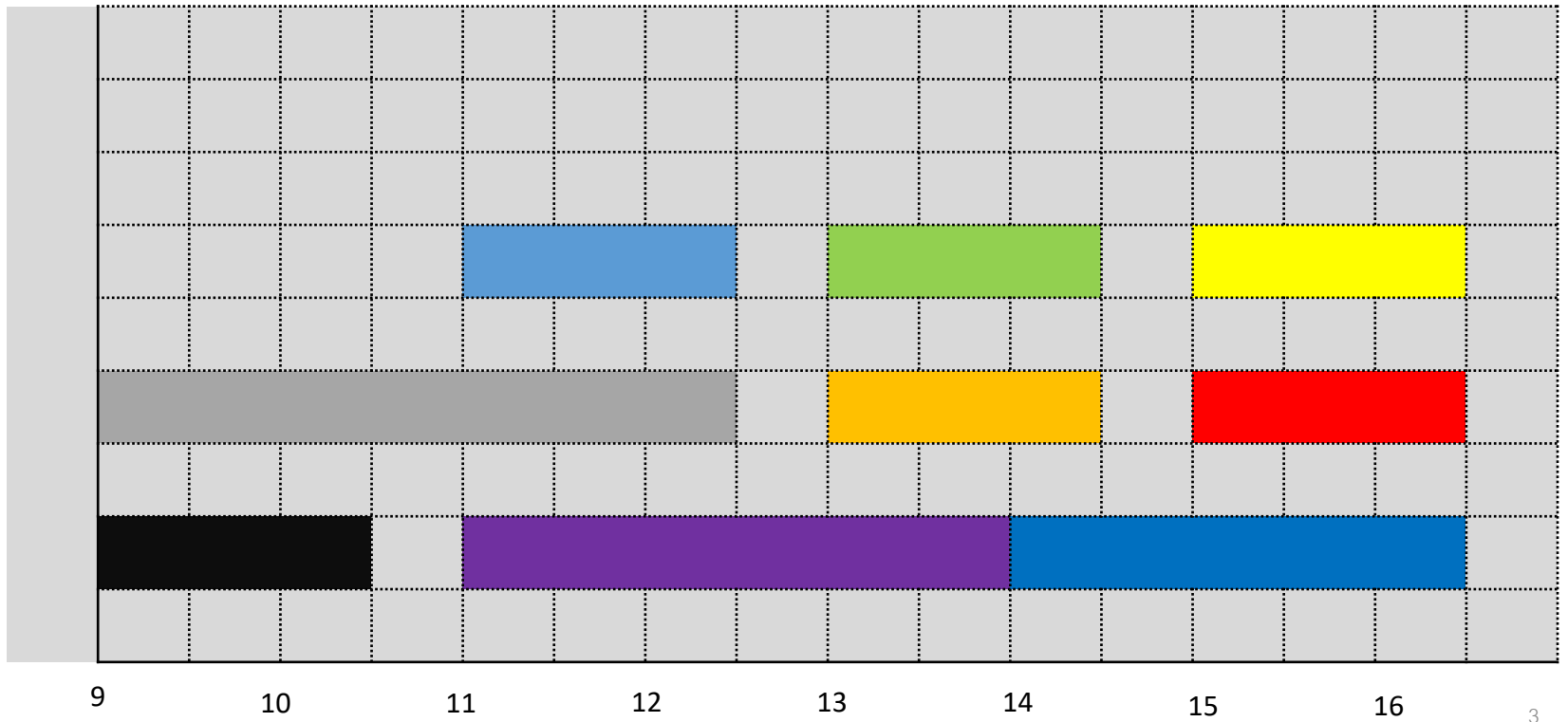
ตัวอย่าง ตารางเวลาแบบนี้ใช้ห้อง 4 ห้องในการจัด 9 รายวิชา



# Interval partitioning

กำหนดให้แต่ละรายวิชา  $j$  มีเวลาเริ่มเรียน  $s_j$  และเวลาเลิกเรียน  $f_j$

ตัวอย่าง ตารางเวลาแบบนี้ใช้ห้อง 3 ห้องในการจัด 9 รายวิชา



# Greedy algorithms

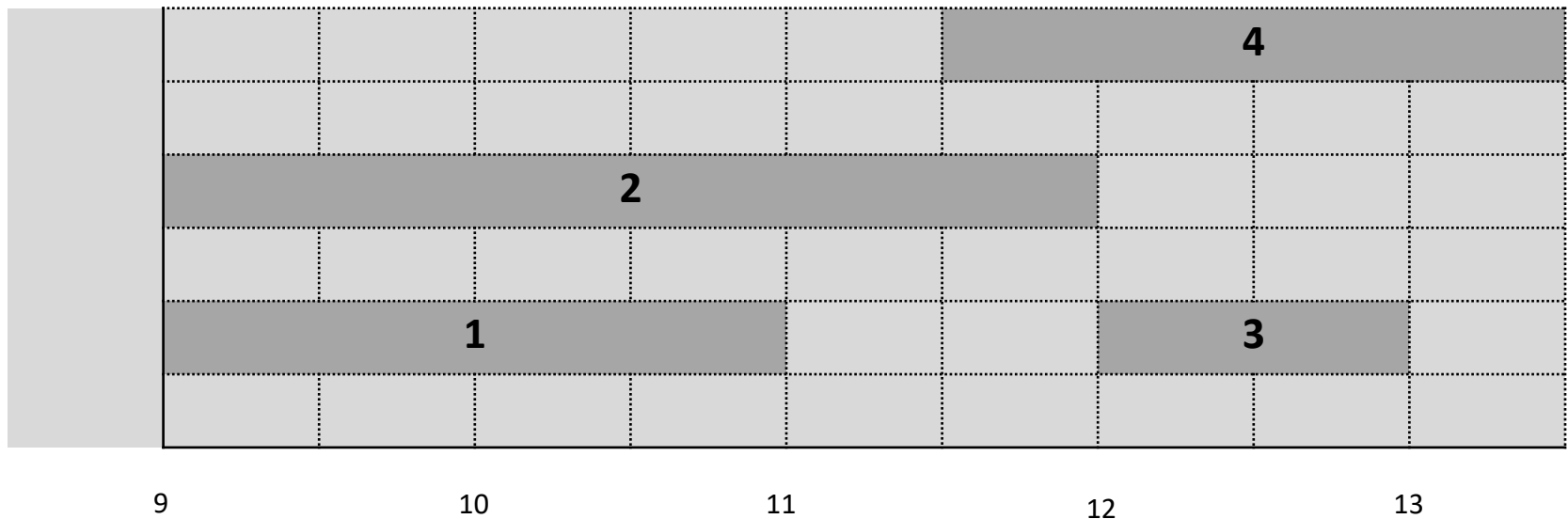
Greedy Template: พิจารณาแต่ละรายวิชาตามลำดับ (ลำดับแบบไหน)

จากนั้นเลือกแต่ละรายวิชาไปใส่ให้กับห้องเรียน (ห้องไหน)

หากไม่มีห้องว่างให้เพิ่มห้อง

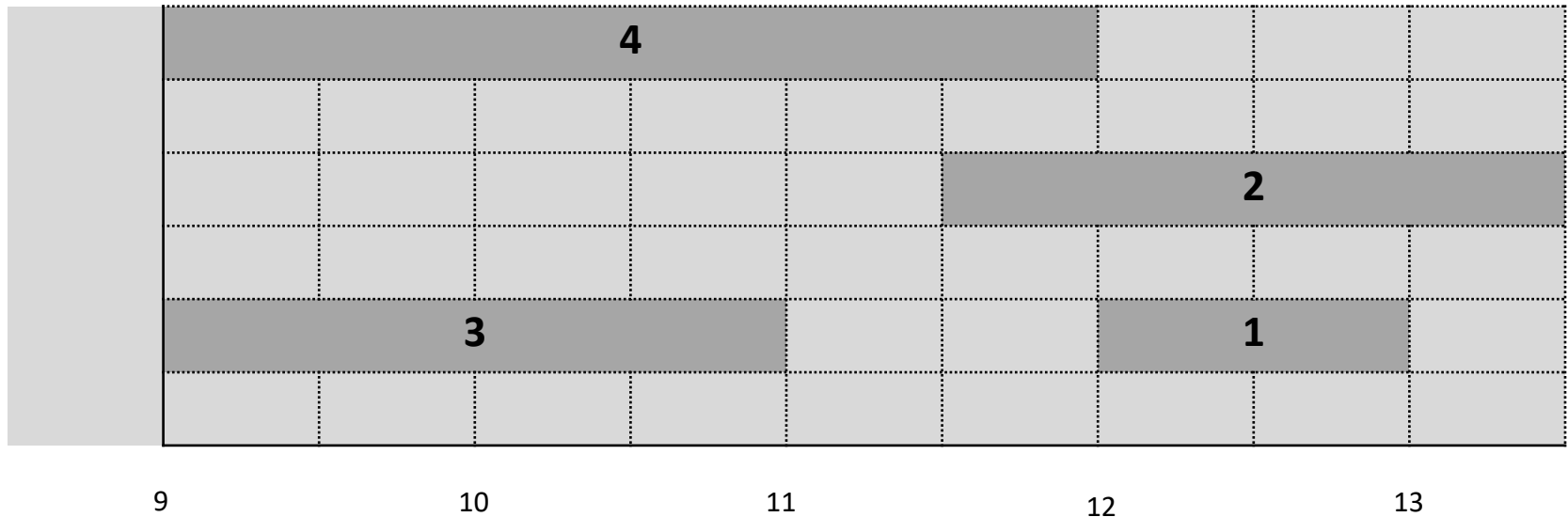
- [เรียงตามเวลาเล็กก่อน] พิจารณารายวิชาตาม  $f_j$  จากน้อยไปมาก
- [เรียงช่วงเวลาที่ใช้น้อยสุดก่อน] พิจารณารายวิชาตาม  $f_j - s_j$  จากน้อยไปมาก
- [นับจำนวนชั้นน้อยสุดก่อน] แต่ละรายวิชา  $j$  นับจำนวนรายวิชาอื่นที่ใช้เวลาร่วมกัน  $c_j$  แล้วจัดตามลำดับ  $c_j$  จากน้อยไปมาก
- [เรียงตามเวลาเริ่มก่อน] พิจารณารายวิชาตาม  $s_j$  จากน้อยไปมาก

# ข้อขัดแย้งกรณี: เรียงตามเวลาเล็กก่อน



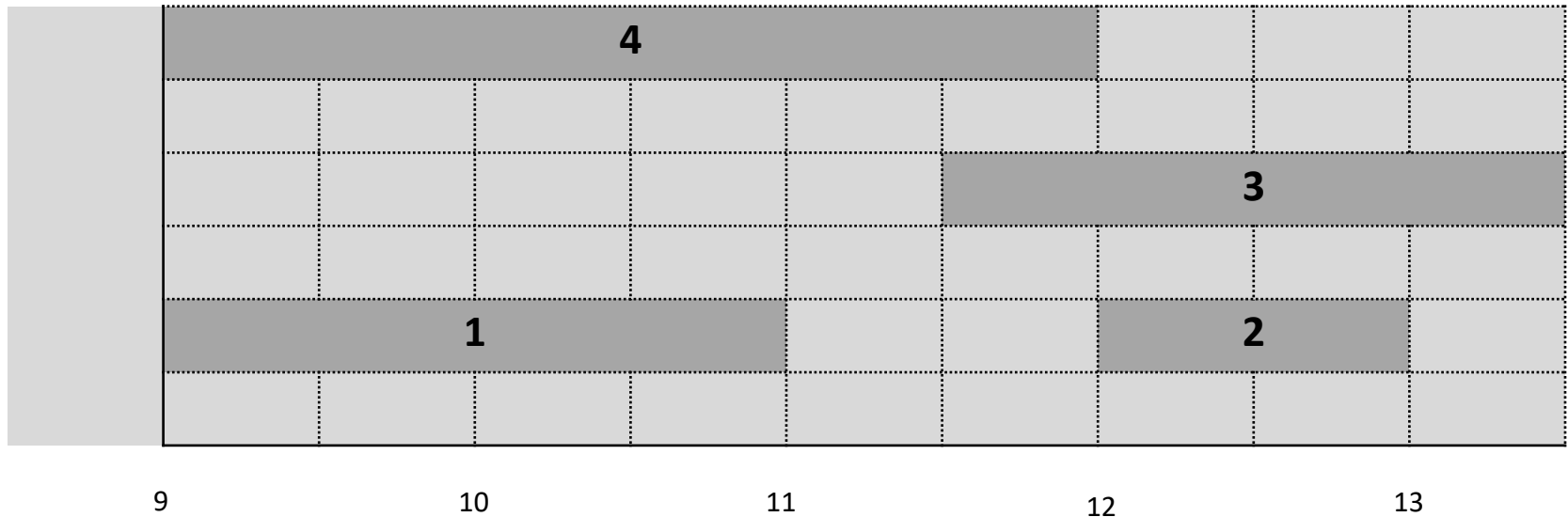
เวลาเรียนของแต่ละรายวิชาเป็นดังนี้ (9,11), (9,12), (11.30,13.30), (12,13)

# ข้อขัดแย้งกรณี: เรียงตามช่วงเวลาที่ใช้น้อยสุดก่อน



เวลาเรียนของแต่ละรายวิชาเป็นดังนี้ (9,11), (9,12), (11.30,13.30), (12,13)

# ข้อขัดแย้งกรณี: เรียงตามจำนวนชั้นน้อยสุดก่อน



เวลาเรียนของแต่ละรายวิชาเป็นดังนี้ (9,11), (9,12), (11.30,13.30), (12,13)

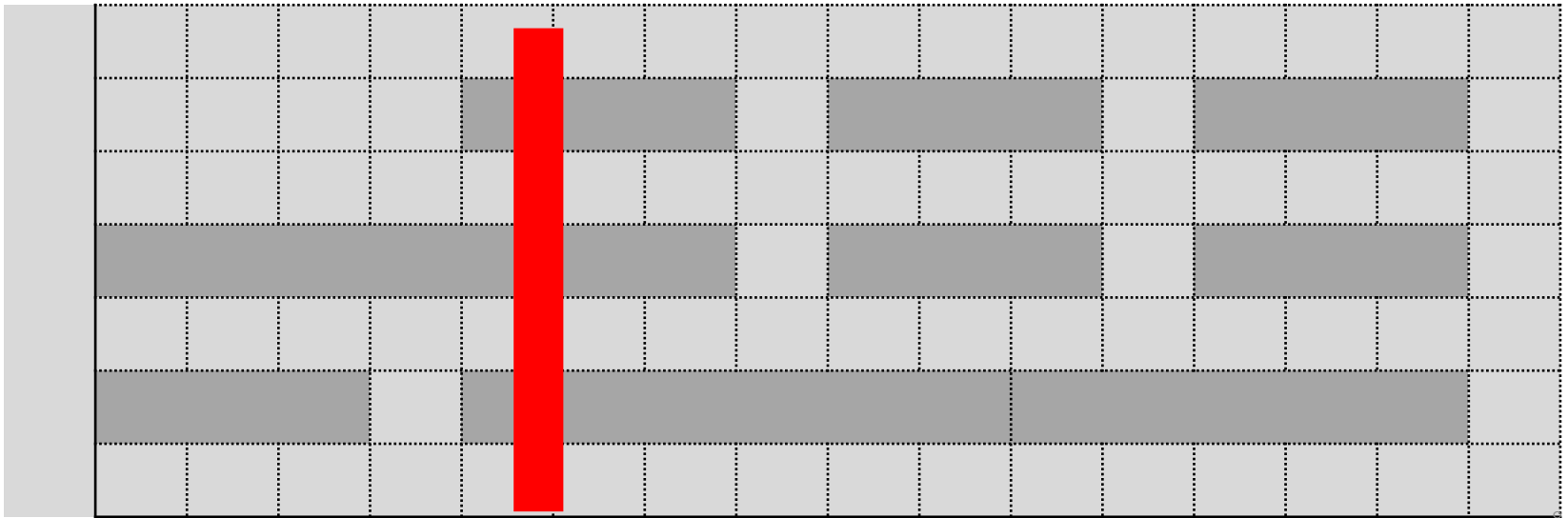
# Lower bound on optimal solution

นิยาม Depth ของเซตของช่วง คือ จำนวนที่มากที่สุดของช่วง(Interval) ณ เวลาใดเวลาหนึ่ง

ข้อสังเกต จำนวนห้องเรียน  $\geq$  Depth

จากตัวอย่างนี้ Depth = 3 ดังนั้น การจัดตารางเรียนแบบนี้ดีที่สุดแล้ว เพราะว่าใช้ 3 ห้อง

Depth = 3





# Lower bound on optimal solution

ข้อสังเกต จำนวนห้องเรียน  $\geq$  Depth

คำถาม มีวิธีการจัดรายวิชาเข้าห้องเรียนที่ใช้ห้องเท่ากับ Depth หรือไม่

Greedy algorithm. พิจารณารายวิชาตามเวลาเริ่ม กำหนดรายวิชาให้กับห้องที่ว่าง หากไม่มีห้องไหนจัดได้เลย ให้เพิ่มห้องใหม่

# Greedy algorithm: Interval partitioning

Sort รายวิชาตามเวลาเริ่ม จะได้ลำดับใหม่ที่  $s_1 < s_2 < \dots < s_n$

$d = 0$  //จำนวนห้อง

for  $j = 1$  to  $n$  {

    if (รายวิชา  $j$  มีห้อง  $k$  ที่ว่าง)

        จัดตารางรายวิชาที่  $j$  ให้กับห้อง  $k$

    else

        เปิดห้องเรียนใหม่  $d + 1$

        จัดตารางรายวิชาที่  $j$  ให้กับห้อง  $d + 1$

$d = d + 1$

}

คำถาม เวลาในการทำงานกรณี Worst case เป็นเท่าไร

คำถาม จะ Implement การหาห้องว่างอย่างไร

# Greedy algorithm: Interval partitioning

Sort รายวิชาตามเวลาเริ่ม จะได้ลำดับใหม่ที่  $s_1 < s_2 < \dots < s_n$

$d = 0$  //จำนวนห้อง

for  $j = 1$  to  $n$  {

    if (รายวิชา  $j$  มีห้อง  $k$  ที่ว่าง)

        จัดตารางรายวิชาที่  $j$  ให้กับห้อง  $k$

    else

        เปิดห้องเรียนใหม่  $d + 1$

        จัดตารางรายวิชาที่  $j$  ให้กับห้อง  $d + 1$

$d = d + 1$

}

Implementation  $O(n \lg n)$

สำหรับแต่ละห้องเรียน  $k$  เราจะเก็บเวลาเลิกสอนของรายวิชาที่เพิ่มไปล่าสุด

เก็บห้องเรียนด้วย priority queue Operation ต่างๆใช้  $O(n \lg n)$

# Analysis of Earliest-start-time-first algorithm

ข้อสังเกต จำนวนห้องเรียน  $\geq$  Depth

ข้อสังเกต Greedy algorithm จะไม่จัดรายวิชาให้ใช้ห้องเดียวกันในเวลาเดียวกัน

ทฤษฎีบท Greedy algorithm ได้คำตอบที่ดีที่สุด

พิสูจน์ วิธีการ Greedy จะจัดห้องให้  $d$  ห้อง ทั้งนี้เราต้องการใช้ห้องอย่างน้อย  $d$

# Analysis of Earliest-start-time-first algorithm

ข้อสังเกต จำนวนห้องเรียน  $\geq$  Depth

ข้อสังเกต Greedy algorithm จะไม่จัดรายวิชาให้ใช้ห้องเดียวกันในเวลาเดียวกัน

ทฤษฎีบท Greedy algorithm ได้คำตอบที่ดีที่สุด

พิสูจน์ วิธีการ Greedy จะจัดห้องให้  $d$  ห้อง ทั้งนี้เราต้องการใช้ห้องอย่างน้อย  $d$

คำถาม ทำไมต้องเปิดห้องที่  $d$

# Analysis of Earliest-start-time-first algorithm

ข้อสังเกต จำนวนห้องเรียน  $\geq$  Depth

ข้อสังเกต Greedy algorithm จะไม่จัดรายวิชาให้ใช้ห้องเดียวกันในเวลาเดียวกัน

ทฤษฎีบท Greedy algorithm ได้คำตอบที่ดีที่สุด

พิสูจน์ วิธีการ Greedy จะจัดห้องให้  $d$  ห้อง ทั้งนี้เราต้องการใช้ห้องอย่างน้อย  $d$  ห้องเรียน  $d$  ถูกเปิดเพราะว่ารายวิชา  $j$  ไม่สามารถจัดให้กับห้อง  $d-1$  ห้องอื่นได้

คำถาม จำนวนห้องที่ไม่สามารถใช้ได้เมื่อเวลา  $s_j + \epsilon$  เป็นเท่าไร (เมื่อ  $\epsilon > 0$  เป็นค่าน้อยๆ)

# Analysis of Earliest-start-time-first algorithm

ข้อสังเกต จำนวนห้องเรียน  $\geq$  Depth

ข้อสังเกต Greedy algorithm จะไม่จัดรายวิชาให้ใช้ห้องเดียวกันในเวลาเดียวกัน

ทฤษฎีบท Greedy algorithm ได้คำตอบที่ดีที่สุด

พิสูจน์ วิธีการ Greedy จะจัดห้องให้  $d$  ห้อง ทั้งนี้เราต้องการใช้ห้องอย่างน้อย  $d$

ห้องเรียน  $d$  ถูกเปิดเพราะว่ารายวิชา  $j$  ไม่สามารถจัดให้กับห้อง  $d-1$  ห้องอื่นได้

เนื่องจากเรา sort ตามเวลาเริ่ม ดังนั้นทุกห้องที่กำหนดให้ไม่ได้นั้นมีรายวิชาที่เริ่มไม่ช้าไปกว่า  $s_j$

ดังนั้น เราจะมี  $d$  รายวิชาที่ซ้อนกันอยู่ ณ เวลา  $s_j + \epsilon$  (เมื่อ  $\epsilon > 0$  เป็นค่าน้อยๆ)

# Analysis of Earliest-start-time-first algorithm

ข้อสังเกต จำนวนห้องเรียน  $\geq$  Depth

ข้อสังเกต Greedy algorithm จะไม่จัดรายวิชาให้ใช้ห้องเดียวกันในเวลาเดียวกัน

ทฤษฎีบท Greedy algorithm ได้คำตอบที่ดีที่สุด

พิสูจน์ วิธีการ Greedy จะจัดห้องให้  $d$  ห้อง ทั้งนี้เราต้องการใช้ห้องอย่างน้อย  $d$

ห้องเรียน  $d$  ถูกเปิดเพราะว่ารายวิชา  $j$  ไม่สามารถจัดให้กับห้อง  $d-1$  ห้องอื่นได้

เนื่องจากเรา sort ตามเวลาเริ่ม ดังนั้นทุกห้องที่กำหนดให้ไม่ได้นั้นมีรายวิชาที่เริ่มไม่ช้าไปกว่า  $s_j$

ดังนั้น เราจะมี  $d$  รายวิชาที่ซ้อนกันอยู่ ณ เวลา  $s_j + \epsilon$  (เมื่อ  $\epsilon > 0$  เป็นค่าน้อยๆ)

จากข้อสังเกตที่ว่าจำนวนห้องเรียน  $\geq$  Depth แสดงว่าทุกๆการจัดนั้นใช้ห้อง  $\geq d$