

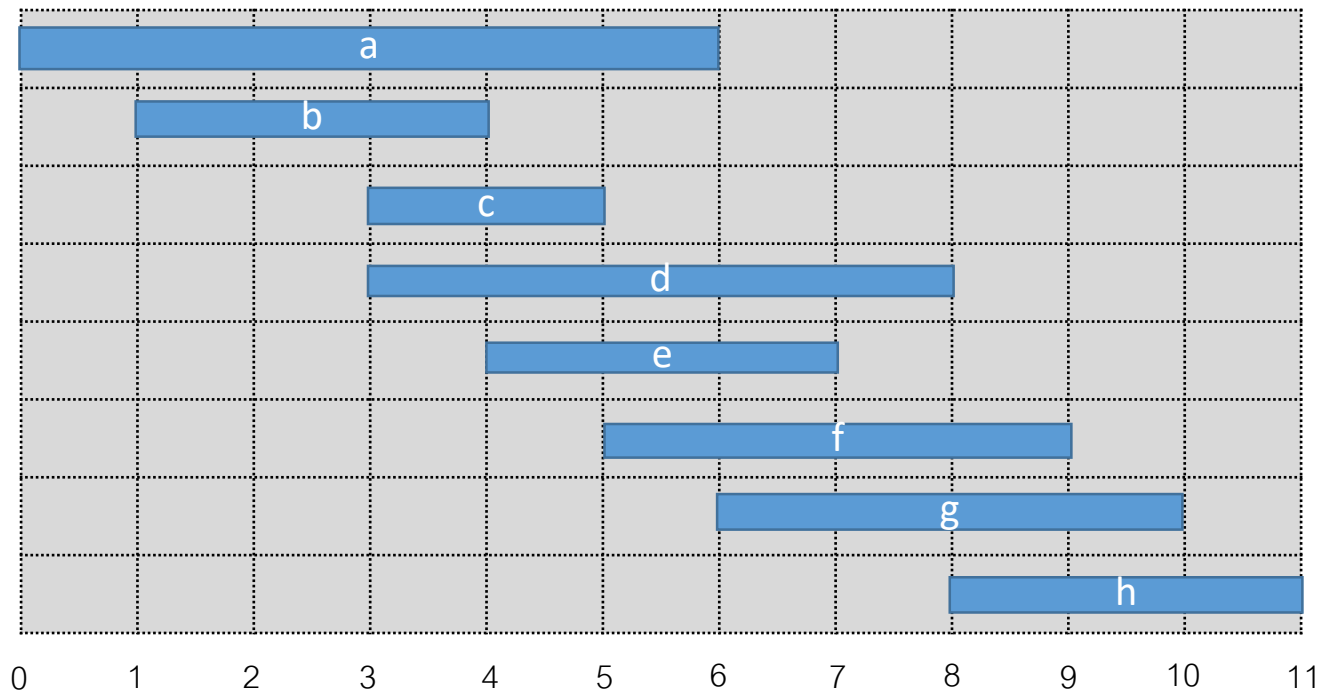
Interval scheduling

Interval scheduling(Activity selection)

กำหนดให้ งาน j เริ่มเวลา s_j และเสร็จเวลา f_j

งานสองงานจะ**สอดคล้องกัน** (compatible) ถ้าเวลาไม่ซ้อนกัน

คำถาม subset ที่ใหญ่ที่สุดของงานที่**สอดคล้องกัน**คืออะไร



Interval scheduling: Greedy algorithms

Greedy Template: พิจารณาแต่ละงานตามลำดับ (ลำดับแบบไหน)

จากนั้นเลือกแต่ละงานหากสอดคล้องกับงานก่อนหน้า

- [เรียงตามเวลาเริ่มก่อน] พิจารณางานตาม s_j จากน้อยไปมาก
- [เรียงตามเวลาเลิกก่อน] พิจารณางานตาม f_j จากน้อยไปมาก
- [เรียงช่วงเวลาที่ใช้น้อยสุดก่อน] พิจารณางานตาม $f_j - s_j$ จากน้อยไปมาก
- [นับจำนวนชิ้นน้อยสุดก่อน] แต่ละงาน j นับจำนวนงานอื่นที่ใช้เวลาร่วมกัน c_j แล้วจัดตามลำดับ c_j จากน้อยไปมาก

- คิดว่าแบบไหนใช้ได้

Interval scheduling: Greedy algorithms

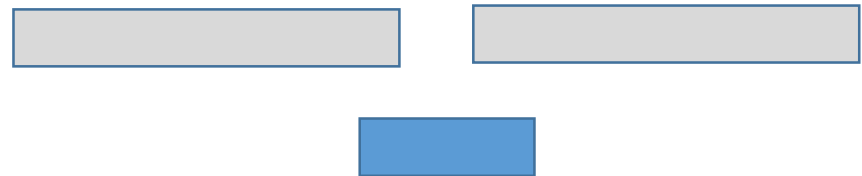
Greedy Template: พิจารณาแต่ละงานตามลำดับ (ลำดับแบบไหน)

จากนั้นเลือกแต่ละงานหากสอดคล้องกับงานก่อนหน้า

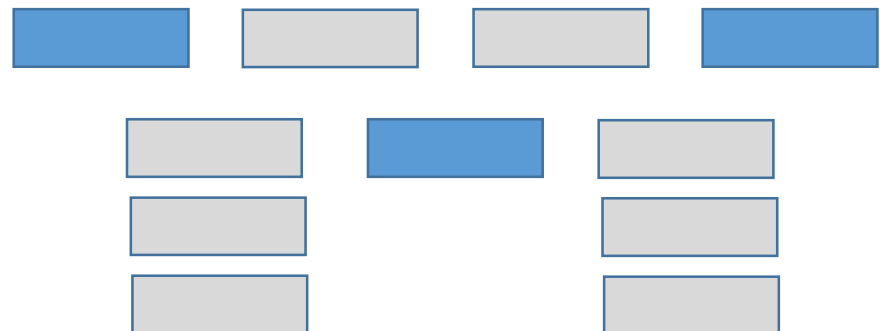
[เรียงตามเวลาเริ่มก่อน]



[เรียงช่วงเวลาที่ใช้น้อยสุดก่อน]



[นับจำนวนชั้นน้อยสุดก่อน]



Interval scheduling: Greedy algorithms

Greedy algorithm: พิจารณางานตามลำดับเวลาเสร็จงานจากน้อยไปมาก เลือกงานที่
สอดคล้องกับงานก่อนหน้า

Sort งานตามลำดับเวลาเสร็จงานจากน้อยไปมาก จะได้ลำดับที่ $f_1 \leq f_2 \leq \dots \leq f_n$

$A = \emptyset$

for $j=1$ to n {

 if(งาน j สอดคล้องกับ A)

$A = A \cup \{j\}$

}

return A

Implement $O(n \log n)$

- งาน j^* ที่เป็นงานที่เพิ่มให้ A งานล่าสุด
- งาน j จะสอดคล้องกับ A ถ้า $s_j \geq f_{j^*}$

Invariant (proof by induction)

บทตั้ง ผลลัพธ์จาก Greedy algorithm นั้นทุกงานสอดคล้องกัน

พิสูจน์ (by induction)

คำถาม ส่วนพื้นฐานของการพิสูจน์แบบ induction มีอะไรบ้าง

Invariant (proof by induction)

บทตั้ง ผลลัพธ์จาก Greedy algorithm นั้นทุกงานสอดคล้องกัน

พิสูจน์ (by induction)

Base: (Initialization)

(Maintenance)

Hypothesis:

Inductive Steps:

Invariant (proof by induction)

บทตั้ง ผลลัพธ์จาก Greedy algorithm นั้นทุกงานสอดคล้องกัน

พิสูจน์ (by induction)

Base: (Initialization) เมื่อ $A = \emptyset$ แล้วทุกงานใน A สอดคล้องกัน

(Maintenance)

Hypothesis: ทุกงาน $i < j$ ใน A สอดคล้องกัน

Inductive Steps: ต้องการพิสูจน์ว่า ทุกงาน $i < j+1$ ใน A สอดคล้องกัน

Invariant (proof by induction)

บทตั้ง ผลลัพธ์จาก Greedy algorithm นั้นทุกงานสอดคล้องกัน

พิสูจน์ (by induction)

Base: (Initialization) เมื่อ $A = \emptyset$ แล้วทุกงานใน A สอดคล้องกัน

(Maintenance)

Hypothesis: ทุกงาน $i < j$ ใน A สอดคล้องกัน

Inductive Steps: ต้องการพิสูจน์ว่า ทุกงาน $i < j+1$ ใน A สอดคล้องกัน

กำหนดให้ ทุกงาน $i < j$ ใน A สอดคล้องกัน

ถ้า $j+1$ ไม่ได้อยู่ใน A แล้วจะได้ว่าทุกงาน $i < j+1$ ใน A สอดคล้องกัน

ถ้าไม่เป็นเช่นนั้น งาน $j+1$ ถูกเพิ่มเข้าไปใน A และเนื่องจากเงื่อนไขของ algorithm ที่จะเพิ่มงาน $j+1$ เมื่อ งาน $j+1$ สอดคล้องกับ A

ดังนั้นทั้งสองกรณีทุกงาน $i < j+1$ ใน A สอดคล้องกัน

สรุป ด้วยการ induction (จนกระทั่ง $j=n$) ทุกงาน ($i < n+1$) ใน A สอดคล้องกัน

Interval scheduling: analysis

ทฤษฎีบท Greedy algorithm ให้ผลลัพธ์ที่ดีที่สุด

พิสูจน์ (โดยข้อขัดแย้ง)

คำถาม เราจะเริ่มพิสูจน์โดยข้อขัดแย้งได้อย่างไร

Proving guide

ในการพิสูจน์ ประพจน์ P โดยข้อขัดแย้ง (contradiction):

1. เขียนบอกว่าเราจะพิสูจน์โดยข้อขัดแย้ง (proof by contradiction)
2. เขียนว่า สมมติว่า P ไม่จริง/ เป็นเท็จ
3. หาข้อสรุปว่าไม่จริง
4. เขียนว่า พบข้อขัดแย้ง ดังนั้น P จะต้องเป็นจริง

Interval scheduling: analysis

ทฤษฎีบท Greedy algorithm ให้ผลลัพธ์ที่ดีที่สุด

พิสูจน์ (โดยข้อขัดแย้ง)

สมมติว่า Greedy **ไม่ได้** ให้ผลลัพธ์ที่ดีที่สุด

Interval scheduling: analysis

ทฤษฎีบท Greedy algorithm ให้ผลลัพธ์ที่ดีที่สุด

พิสูจน์ (โดยข้อขัดแย้ง)

สมมติว่า Greedy ไม่ได้ให้ผลลัพธ์ที่ดีที่สุด

คำถาม เราจะทำอย่างไรเพื่อให้ไปถึงข้อขัดแย้ง

คำตอบ ดูว่าจุดไหนที่คำตอบที่ดีที่สุดต่างจาก Greedy

Interval scheduling: analysis

ทฤษฎีบท Greedy algorithm ให้ผลลัพธ์ที่ดีที่สุด

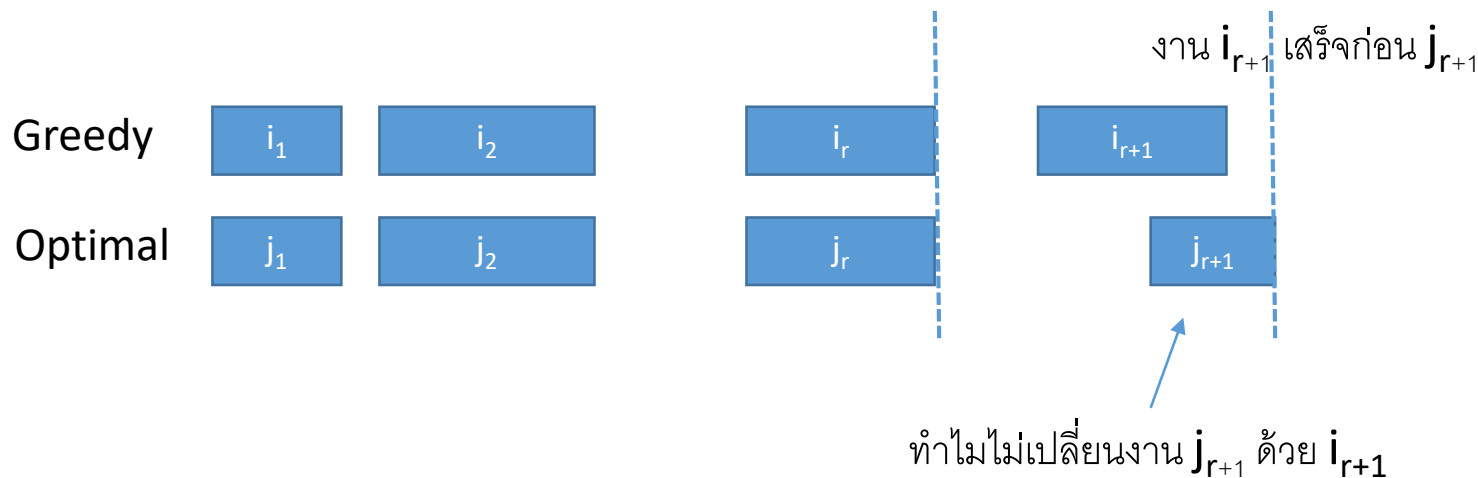
พิสูจน์ (โดยข้อขัดแย้ง)

สมมติว่า Greedy ไม่ได้ให้ผลลัพธ์ที่ดีที่สุด

ให้ i_1, i_2, \dots, i_k แทนเซตของงานที่ถูกเลือกด้วยวิธี greedy

ให้ j_1, j_2, \dots, j_k แทนเซตของงานในคำตอบที่ดีที่สุด

พิจารณาคำตอบที่ดีที่สุด (OPT) ที่เลือกงานเหมือนกับ Greedy ตั้งแต่เริ่มต้นจนถึงจุดที่ต่าง (จนถึง r) นั่นคือ $i_1=j_1, i_2=j_2, \dots, i_r=j_r$ สำหรับค่า r ที่มากที่สุดที่เป็นไปได้



Interval scheduling: analysis

ทฤษฎีบท Greedy algorithm ให้ผลลัพธ์ที่ดีที่สุด

พิสูจน์ (โดยข้อขัดแย้ง)

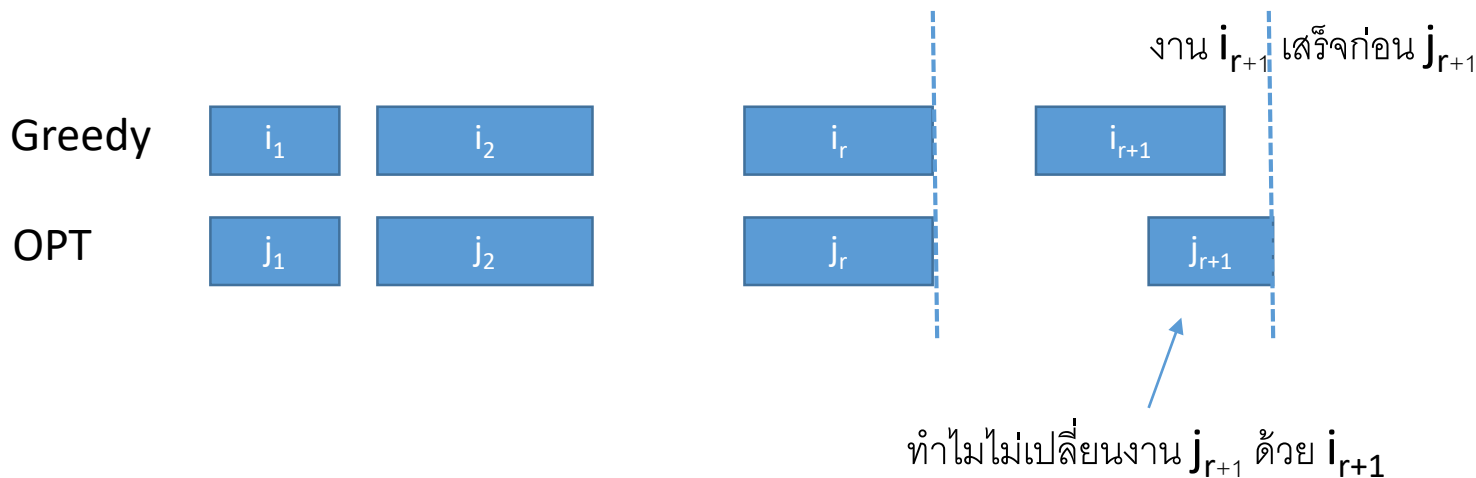
สมมติว่า Greedy ไม่ได้ให้ผลลัพธ์ที่ดีที่สุด

ให้ i_1, i_2, \dots, i_k แทนเซตของงานที่ถูกเลือกด้วย greedy

ให้ j_1, j_2, \dots, j_k แทนเซตของงานในคำตอบที่ดีที่สุด

พิจารณาคำตอบที่ดีที่สุด (OPT) ที่เลือกงานเหมือนกับ Greedy ตั้งแต่เริ่มต้นจนถึงจุดที่ต่าง (จนถึง r) นั่นคือ $i_1 = j_1, i_2 = j_2, \dots, i_r = j_r$ สำหรับค่า r ที่มากที่สุดที่เป็นไปได้

คำถาม ขัดแย้งตรงไหน



Interval scheduling: analysis

ทฤษฎีบท Greedy algorithm ให้ผลลัพธ์ที่ดีที่สุด

พิสูจน์ (โดยข้อขัดแย้ง)

สมมติว่า Greedy ไม่ได้ให้ผลลัพธ์ที่ดีที่สุด

ให้ i_1, i_2, \dots, i_k แทนเซตของงานที่ถูกเลือกด้วย greedy

ให้ j_1, j_2, \dots, j_k แทนเซตของงานในคำตอบที่ดีที่สุด

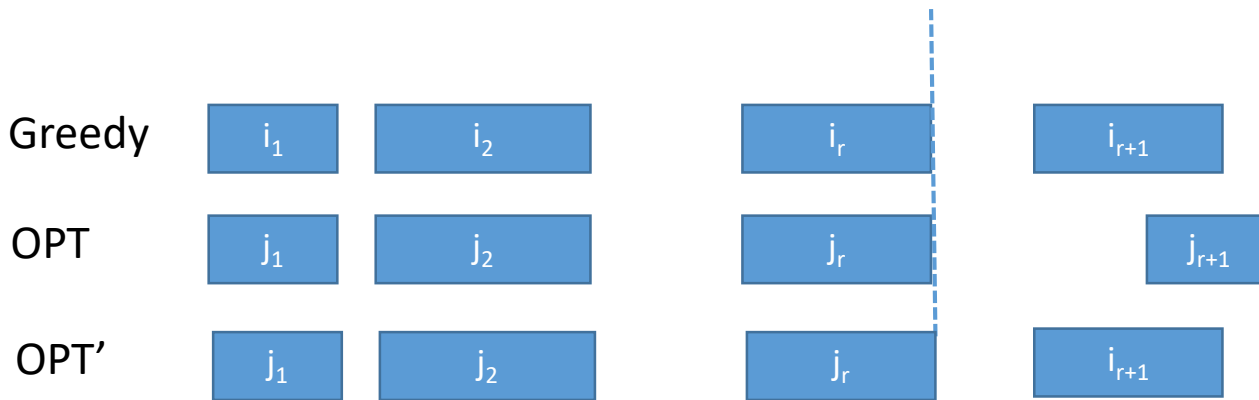
พิจารณาคำตอบที่ดีที่สุด (OPT) ที่เลือกงานเหมือนกับ Greedy ตั้งแต่เริ่มต้นจนถึงจุดที่ต่าง (จนถึง r) นั่นคือ $i_1 = j_1, i_2 = j_2, \dots, i_r = j_r$ สำหรับค่า r ที่มากที่สุดที่เป็นไปได้

พิจารณาคำตอบที่ต่างกัน

เราสามารถเปลี่ยน OPT ไปเป็น OPT' ซึ่งยังคงเป็นคำตอบที่ดีที่สุดแต่เลือกงานเหมือนกับ Greedy ที่มีความยาวมากกว่าเดิม

ขัดแย้ง: OPT' เลือกงานเหมือน Greedy ที่ยาวกว่า OPT

(คำตอบใหม่ OPT' ยังคงเป็นคำตอบที่เป็นไปได้และเป็นคำตอบที่ดีที่สุด แต่ขัดแย้งตรงที่บอกว่าเราเลือก opt ที่ r มากสุดแล้ว)



- สังเกตว่า OPT' ก็ยังคงเป็น optimal solution
- แต่มีส่วนที่เหมือนกับ greedy มากกว่าเดิม
- ขัดแย้งที่บอกว่าเราเลือก OPT ที่เหมือนกับ greedy ยาวที่สุดแล้ว

Greedy stays ahead

บทตั้ง สำหรับทุกๆ $r \leq k$, $f(i_r) \leq f(j_r)$ เมื่อ i เป็นงานที่ Greedy เลือกและ j เป็นงานที่ OPT เลือก

พิสูจน์ (by induction:)

คำถาม ส่วนพื้นฐานของการพิสูจน์แบบ induction มีอะไรบ้าง

Greedy stays ahead

บทตั้ง สำหรับทุกๆ $r \leq k$, $f(i_r) \leq f(j_r)$ เมื่อ i เป็นงานที่ Greedy เลือก j OPT เลือก

พิสูจน์ (by induction:)

Base เมื่อ $k=1$

Hypothesis: สมมติว่า สำหรับทุกๆ $r \leq k$, $f(i_r) \leq f(j_r)$

Inductive Steps: ต้องการพิสูจน์ว่า สำหรับทุกๆ $r \leq k+1$, $f(i_r) \leq f(j_r)$

Greedy stays ahead

บทตั้ง สำหรับทุกๆ $r \leq k$, $f(i_r) \leq f(j_r)$ เมื่อ i เป็นงานที่ Greedy เลือก j OPT เลือก

พิสูจน์ (by induction:)

Base เมื่อ $k=1$, กรณีที่เลือกงานแรก i_1 จะถูกเลือกโดยที่ $f(i_1) \leq f(j_1)$

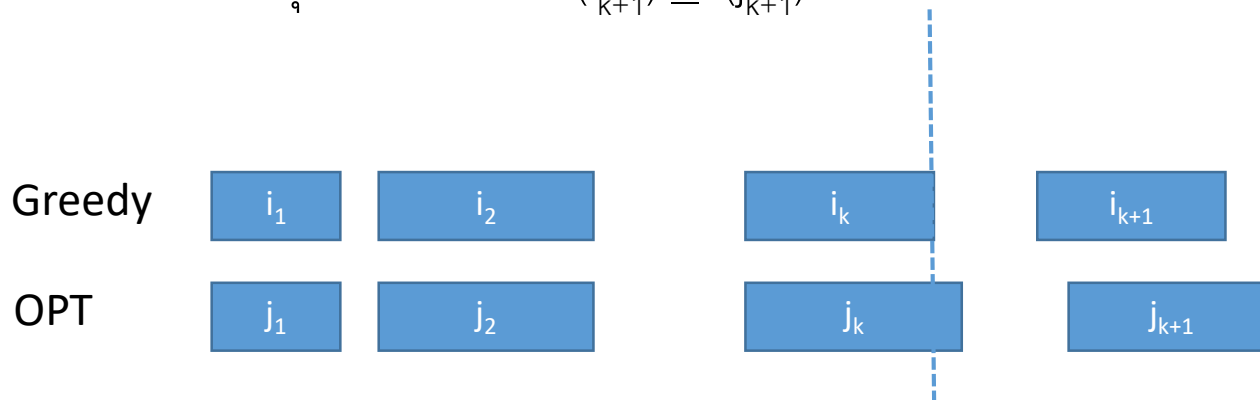
Hypothesis: สมมติว่า สำหรับทุกๆ $r \leq k$, $f(i_r) \leq f(j_r)$

Inductive Steps: ต้องการพิสูจน์ว่า สำหรับทุกๆ $r \leq k+1$, $f(i_r) \leq f(j_r)$

สำหรับทุกๆ $r \leq k$ จะเป็นจริงตามสมมติฐาน

พิจารณา $r=k+1$

คำถาม เราจะสรุปได้อย่างไรว่า $f(i_{k+1}) \leq f(j_{k+1})$



Greedy stays ahead

บทตั้ง สำหรับทุกๆ $r \leq k$, $f(i_r) \leq f(j_r)$ เมื่อ i เป็นงานที่ Greedy เลือก j OPT เลือก

พิสูจน์ (by induction:)

Base เมื่อ $k=1$, กรณีที่เลือกงานแรก i_1 จะถูกเลือกโดยที่ $f(i_1) \leq f(j_1)$

Hypothesis: สมมติว่า สำหรับทุกๆ $r \leq k$, $f(i_r) \leq f(j_r)$

Inductive Steps: ต้องการพิสูจน์ว่า สำหรับทุกๆ $r \leq k+1$, $f(i_r) \leq f(j_r)$

สำหรับทุกๆ $r \leq k$ จะเป็นจริงตามสมมติฐาน

พิจารณา $r=k+1$

คำถาม เราจะสรุปได้อย่างไรว่า $f(i_{k+1}) \leq f(j_{k+1})$

คำถาม Greedy เลือกงานชิ้นไหน



Greedy stays ahead

บทตั้ง สำหรับทุกๆ $r \leq k$, $f(i_r) \leq f(j_r)$ เมื่อ i เป็นงานที่ Greedy เลือก j OPT เลือก

พิสูจน์ (by induction:)

Base เมื่อ $k=1$, กรณีที่เลือกงานแรก i_1 จะถูกเลือกโดยที่ $f(i_1) \leq f(j_1)$

Hypothesis: สมมติว่า สำหรับทุกๆ $r \leq k$, $f(i_r) \leq f(j_r)$

Inductive Steps: ต้องการพิสูจน์ว่า สำหรับทุกๆ $r \leq k+1$, $f(i_r) \leq f(j_r)$

สำหรับทุกๆ $r \leq k$ จะเป็นจริงตามสมมติฐาน

พิจารณา $r=k+1$

เรารู้ว่า $f(j_k) \leq s(j_{k+1})$ จาก OPT

ดังนั้น $f(i_k) \leq s(j_{k+1})$ จากสมมติฐาน

ทำให้งาน j_{k+1} สามารถถูกเลือกได้จาก Greedy (i_{k+1} สามารถมีค่าเท่ากับ j_{k+1} ได้)

เนื่องจาก Greedy เลือกงานที่มีเวลาเสร็จเร็วที่สุด ดังนั้น $f(i_{k+1}) \leq f(j_{k+1})$

Interval scheduling: analysis

ทฤษฎีบท Greedy algorithm ให้ผลลัพธ์ที่ดีที่สุด

พิสูจน์ (by contradiction)

ให้ i_1, i_2, \dots, i_k แทนเซตของงานที่ถูกเลือกด้วย greedy

ให้ j_1, j_2, \dots, j_m แทนเซตของงานในคำตอบที่ดีที่สุด

สมมติว่า Greedy ไม่ได้ให้คำตอบที่ดีที่สุด ดังนั้น $k < m$

อย่างไรก็ตามสำหรับทุกๆ $r \leq k$, $f(i_r) \leq f(j_r)$ จากบทตั้งก่อนหน้า

ดังนั้น $f(i_k) \leq f(j_k)$

แต่ยังมีงาน j_{k+1} ที่เลือกได้ซึ่งเริ่มหลังจาก j_k นั่นคือหลังจาก i_k เสร็จแล้ว

แต่หลังจากที่ Greedy เลือก i_k ยังมีงานที่สอดคล้องเหลืออยู่อีก

ขัดแย้งกับที่ว่า การเลือกด้วยวิธี Greedy มีแค่ k งาน