

Algorithm Design and Analysis

บทที่ 10
Network flow Part II

Greedy

หา st-path ที่แต่ละเส้นเชื่อมมี $Flow f(e) \leq Capacity c(e)$
จากนั้น augment flow ตาม path นั้น

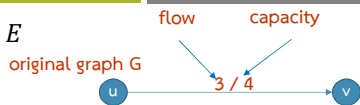
ลักษณะการทำงานของ Greedy คือ ทำซ้ำจนหา path เพิ่มไม่ได้

ข้อเสียคือ การที่จะทำให้ได้ดีที่สุดต้อง backtrack

Residual graph

Original edge: $e = (u, v) \in E$

- Flow $f(e)$
- Capacity $c(e)$

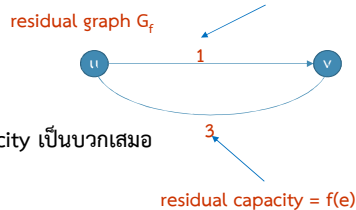


Residual edge: เพิ่มเส้นย้อนกลับกับ flow ที่ถูกส่ง residual capacity = $c(e) - f(e)$

- $e = (u, v)$ และ $e^R = (v, u)$

Residual graph:

- ▶ ทุกเส้นเชื่อมจะมี residual capacity เป็นบวกเสมอ

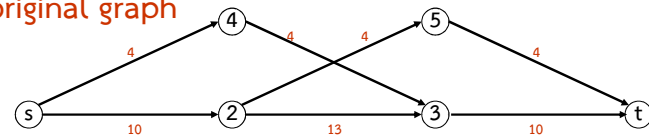


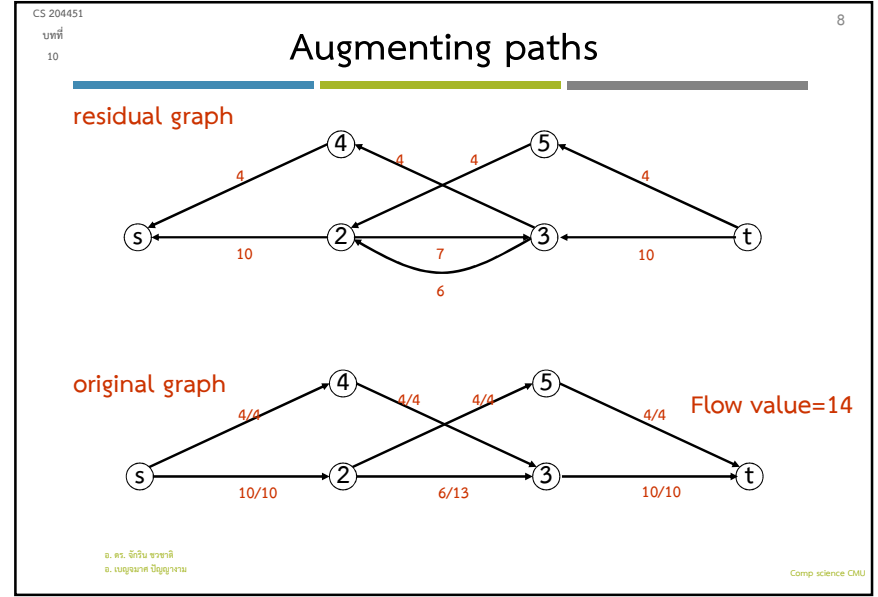
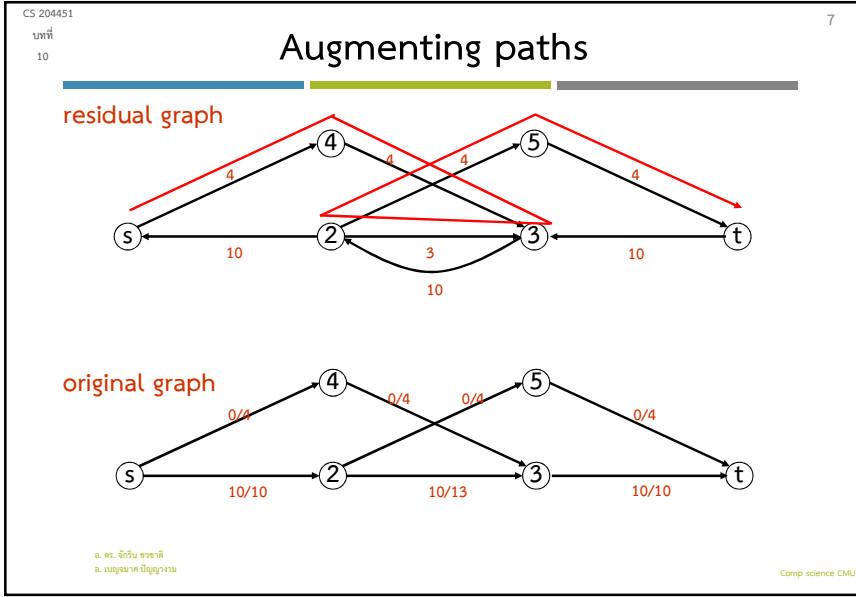
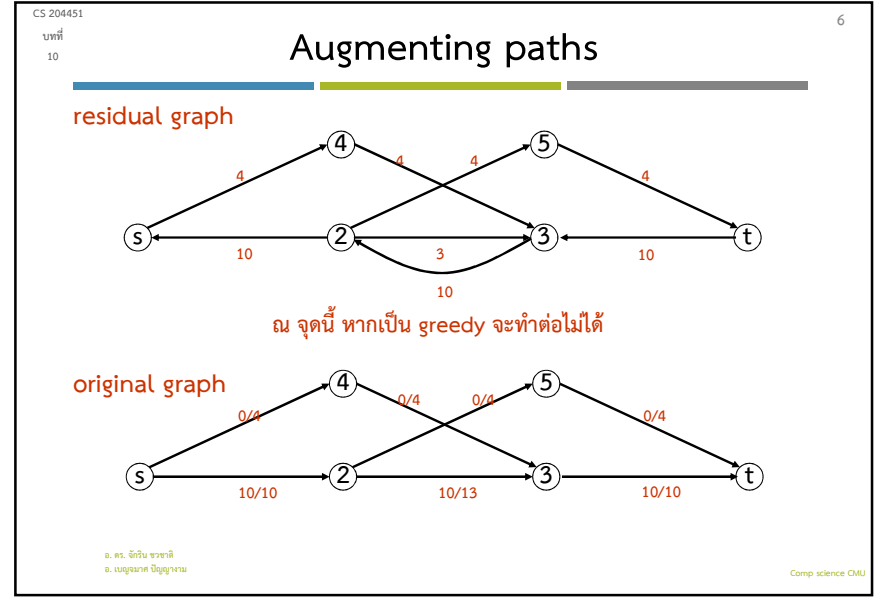
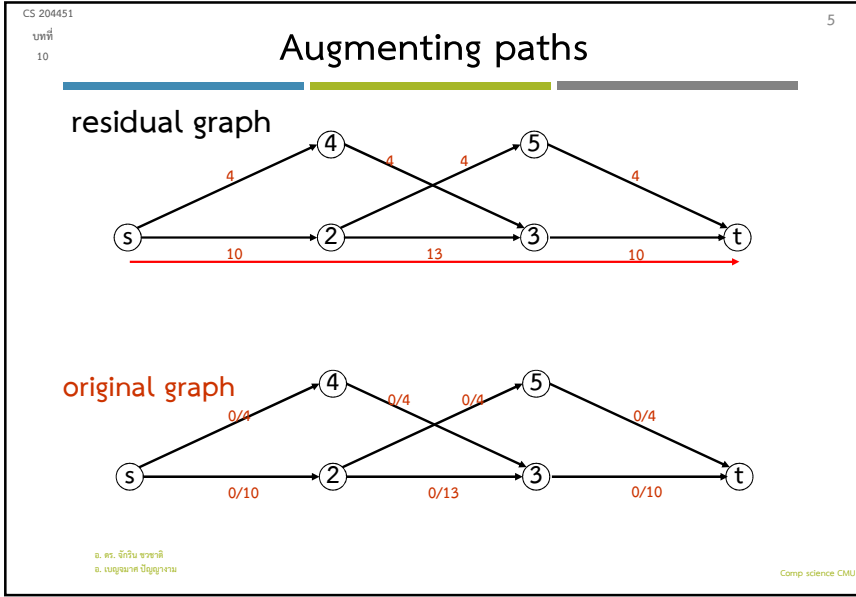
Augmenting paths

Augmenting path คือ st-path ใน residual graph

- จะเพิ่ม flow ไปตาม forward edges (ตามทิศทางของเส้นเชื่อม)
- จะลด flow ไปตาม backward edges (ตามทิศย้อนเส้นเชื่อม)

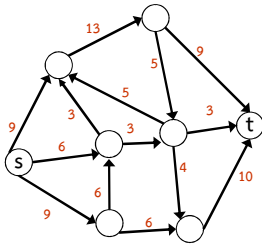
original graph





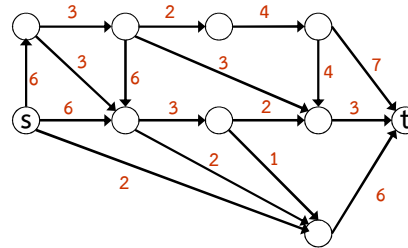
Assignment ครั้งที่ 6 ข้อ 1

- ให้หาค่า max flow ที่ได้ โดยวิธีการหา Augmenting paths และการสร้าง Residual graph (ให้แสดงวิธีหา Mincut ของโจทย์ทั้งสองข้อนี้ด้วย)



Assignment ครั้งที่ 6 ข้อ 2

- ให้หาค่า max flow ที่ได้ โดยวิธีการหา Augmenting paths และการสร้าง Residual graph (ให้แสดงวิธีหา Mincut ของโจทย์ทั้งสองข้อนี้ด้วย)



Ford-Fulkerson Augmenting Path Algorithm

```

while(there exists an augmenting path){
    Find augmenting path P
    Compute bottleneck capacity of P
    Augment flow along P
}

```

คำถาม

Q: การทำเช่นนี้ทำให้ได้ max flow ใช่หรือไม่

A: ใช่

Ford-Fulkerson Algorithm: Analysis

Assumption: ให้ความจุมีค่าเป็นจำนวนเต็ม

ในแต่ละรอบการทำงานจะมีการหา st-path ซึ่งใช้เวลาในการทำงาน $O(N+M)$

เราจะส่ง flow อย่างน้อย 1 หน่วยผ่าน path นี้

ถ้า max-flow มีค่าเป็น f^* แล้วเวลาในการทำงานของ อัลกอริทึมมีค่าเป็น $O((N+M)*|f^*|)$

CS 204451
บทที่ 10

Choosing Good Augmenting Path

residual graph

original graph

อ. ดร. จักรีน ซาซาลี
อ. บุญจนาต ปัญญาธรรม

Comp science CMU

CS 204451
บทที่ 10

Choosing Good Augmenting Path

residual graph

original graph

อ. ดร. จักรีน ซาซาลี
อ. บุญจนาต ปัญญาธรรม

Comp science CMU

CS 204451
บทที่ 10

Choosing Good Augmenting Path

residual graph

original graph

อ. ดร. จักรีน ซาซาลี
อ. บุญจนาต ปัญญาธรรม

Comp science CMU

CS 204451
บทที่ 10

Choosing Good Augmenting Path

residual graph

original graph

หากเลือกไม่ได้ต้องทำถึง 200 รอบ

อ. ดร. จักรีน ซาซาลี
อ. บุญจนาต ปัญญาธรรม

Comp science CMU

Choosing Good Augmenting Path

17

ควรระวังในการเลือก augmenting path

- ▶ การเลือกบางวิธีทำให้ได้ exponential algorithm
- ▶ การเลือกบางวิธีทำให้ได้ polynomial algorithm

ออกแบบรูปแบบของ augmenting path ที่อยากได้

- ▶ หา augmenting path ได้อย่างมีประสิทธิภาพ
- ▶ ทำจำนวนรอบที่น้อย

เลือก augmenting path ด้วยวิธี

- ▶ ใช้จำนวนเส้นเชื่อมที่น้อยที่สุด (shortest path)
- ▶ เลือกเส้นที่ bottleneck capacity ใหญ่สุด (fattest path)

Shortest augmenting path

18

หาได้ง่าย สามารถใช้ BFS ได้

หา augmenting path ที่มีจำนวนเส้นเชื่อมน้อยที่สุด

ในแต่ละรอบความยาวของ shortest augmenting path จะเพิ่มขึ้น

- ▶ ความยาวเพิ่มไม่เกิน E
- ▶ ไม่เกิน EV augmenting path ทั้งหมด
- ▶ ดังนั้นเวลาในการทำงานเป็น $O(E^2V)$

Fattest augmenting path

19

หา augmenting path ที่มี bottleneck capacity ที่มีค่ามากที่สุด

ส่ง flow ไปยัง sink

solve โดยใช้ dijkstra-style (Priority-first search) algorithm

การหา fattest path ใช้ $O(E \log V)$ ต่อการ augment โดยใช้ binary heap

การหา min-cut

20

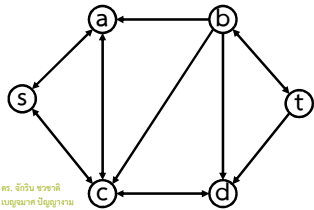
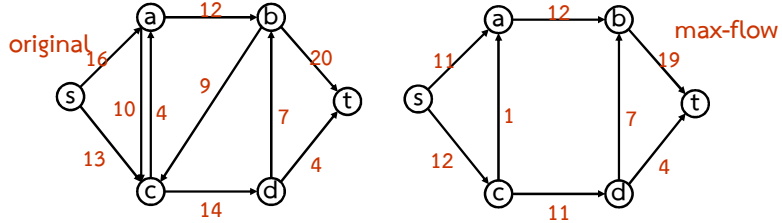
- เรารู้ว่า max flow – min cut
- ตอนนี้เรารู้วิธีการหา max flow

Q: เราจะหา min-cut ได้อย่างไร

A: เราจะใช้ residual graph

การทำ min-cut

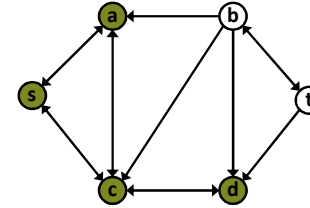
- เราจะลบ max flow ออกจาก original graph



แสดงแค่ topology ของ residual graph
อย่าลืมเพิ่มเส้นย้อนกลับ

การทำ min-cut จาก topology ของ residual graph

- mark ทุกโหนดที่ไปถึงจาก s
 - เรียก set ของโหนดที่ไปถึงจาก s ว่า A



แยกโหนดเหล่านี้ออกจากกลุ่ม
เส้นเชื่อมที่วิ่งจาก A ไปยัง V-A คือ cut นั้นเอง

การทำ min-cut

- พิจารณาใน original graph เพื่อให้ cut

