

DAY 2



ADVANCE GRAPH

เฉลยการบ้าน ทางไก่



combinatorics แล้วก็วนหา

$$\text{num}[i] * \text{num}[M-i]$$

เฉลยการบ้าน TUMSO18 MATHMATH



$$\frac{10^{126k+3} + 143}{127}$$

สำหรับ $k = 1$

$$\frac{10^{126 \cdot 1 + 3} + 143}{127} = 7874015748031496062992125984251968503937007874015748031496062992125984251968503937007874015748031496062992125984251968503937009$$

787401574803149606299212598425196850393700	189	
787401574803149606299212598425196850393700	189	(567 * 1) + 9
787401574803149606299212598425196850393700	189	
9	9	

K=2	7 874 015 748 031 496 062 992 125 984 251 968 503 937 00	189	(567 * 2) + 9
	7 874 015 748 031 496 062 992 125 984 251 968 503 937 00	189	
	7 874 015 748 031 496 062 992 125 984 251 968 503 937 00	189	
	7 874 015 748 031 496 062 992 125 984 251 968 503 937 00	189	
	7 874 015 748 031 496 062 992 125 984 251 968 503 937 00	189	
	7 874 015 748 031 496 062 992 125 984 251 968 503 937 00	189	
	9	9	

$(567 * n) + 9$ นั่นเอง

จุดผิดของข้อนี้คือเลขทศนิยมมากเกินไป เขียนโปรแกรม observe จะได้ค่าไม่ตรง ให้ใช้เว็บนี้แทน
<https://defuse.ca/big-number-calculator.htm>

เฉลยการบ้าน CSES FERRIS WHEEL



ทางที่ดีคือ ให้จับเด็กบน.มาและบน.น้อยมานั่งด้วย
sort แล้ว ใช้ two pointer จากต้นและปลาย เขยิบๆเลือกเด็ก



กราฟที่มี BITMARK

กราฟเก็บ STAGE



กราฟที่ระหว่างเดินกราฟ เราสามารถทำอะไรบ้างอย่างที่ใช้แล้วหมดไป/ต้องรวบรวม ได้ เช่น

- สามารถเลือก path ที่ไม่เสียค่า w ได้ $[k]$ ครั้ง
- ต้องเดินทางไปเก็บสิ่งของ k ลักษณะระหว่างทาง

การเก็บ $depth[N]$ จะไม่พอเพราะมันมีหลาย stage \rightarrow เราจะใช้ array 2 มิติ $depth[2^{*(k-1)}][N]$ แทน

ส่วนฟังก์ชัน เราจะเขียน $DFS(v, w, \text{ค่า stage})$

ไปแบบปกติ $DFS(v, w+ww, 0)$ ต่อ



ถ้ายังไม่ใช้ stage : สามารถ ไป $DFS(v, w, 1)$ ได้



Equipped [20 min]

เฉลย EQUIPPED



เราใช้ $dp[2^{**}(k)]$ ช่องเก็บว่าแต่ละช่องใช้ค่าใช้จ่ายน้อยสุดเท่าไหร่ [ใช้ $1 \leq K$ เป็นการขยับบิต ได้ $2^{**}K$]

task 1 : รับของมาแปลงเป็น index ของช่อง

0000 = 0 , 0001 = 1, 0010 = 2, 0011 = 3 ,

task 2 : วน K เพื่อเลือกสิ่งของที่จะเอามารวมกัน เช่น หาก i และ j ทำได้ ระหว่างเลือก i,j ใหม่ กับใช้ของเดิม อะไรดีหว่ากัน

$dp[i|j] = \min(dp[i|j], dp[i] + dp[j])$



magic pooh [20 min]

เฉลย MAGIC POOH



เป็น กราฟที่เก็บ stage ระหว่าง "ใช้ไม่ไฟฟ์ไปแล้ว" กับ "ยังไม่ได้ใช้ไม่ไฟฟ์"
เราจะใช้ array 2 มิติ แทนปกติที่ใช้ 1 มิติ เพราะจะมีช่อง ใช้ไปแล้วเพิ่มมา

ถ้ายังไม่ใช้ไม่ไฟฟ์ พอถึง node ใดๆ เราจะแตกต่างระหว่างใช้/ไม่ใช้ลงไป priority queue
เนื่องจากเป็น pqueue ดังนั้นมันพิจารณาทางที่ w น้อยที่สุดมาก่อนอยู่แล้ว เราเลยแตกรู้ๆได้เลย



Seven Gems [30 min]

เจดย SEVEN GEMS



N, M = 200 ดูไม่เยอะ แต่การที่มีหลาย GEM นี่แหละ จะทำไงให้ไปได้ดีที่สุด

ทำแบบเดิม คือ ใช้ `depth[N][M]` [7] [2**7]

จำนวน gem ที่เก็บได้ เก็บ GEM ไหนไปแล้วบ้าง

เราจะให้ Gem แต่ละอันต่างกัน -> gem1 gem2 gem3

stage มันเยอะนะ อาจใช้ struct หรือ tuple แทน

```
#define tuu tuple<int, int, int, int, int, int>
```

```
x, y, time, time_m (mod 7), gem_mask, gem_num
```

direction มี 5 แบบ คือ ขึ้น-ลง-ซ้าย-ขวา-อยู่ที่เดิม (อยู่ที่เดิมแค่ครบ 6 5)

ตรง time mod อาจใช้การค่อยๆบวก แล้วปัดลงเป็น 1 ทุกครั้งที่ถึง 7 ได้



หา NODE ก่อนหน้า NODE



วิธีสิ้นสุดแบบแหล่งต้นทางเดียว
[20 min]

เฉลย วิธีสั้นสุดแบบแหล่งต้นทางเดียว



เนื่องจากระยะทางคือผลคูณของสองจำนวนเฉพาะ หากเรา คูณตรงๆมันจะเยอะมากๆ
ใช้คุณสมบัติ $\text{LOG}_{10}(W)$ แทน ใช้ได้เพราะระยะทางมันไม่ถูดยกมาใช้อีกรอบอยู่ละ แค่เทียบว่ามาก-น้อยกว่าก็พอ

POINT : DFS หาเส้นทางที่สั้น แล้ว ย้อนเส้นทางที่เคยผ่านมา

ตอนDFS ให้ MARK ไว้ว่า U ที่มาหา V เป็นใคร แล้วตอนย้อนก็ย้อนไปเรื่อยๆจนกว่า $PA[U] == U$

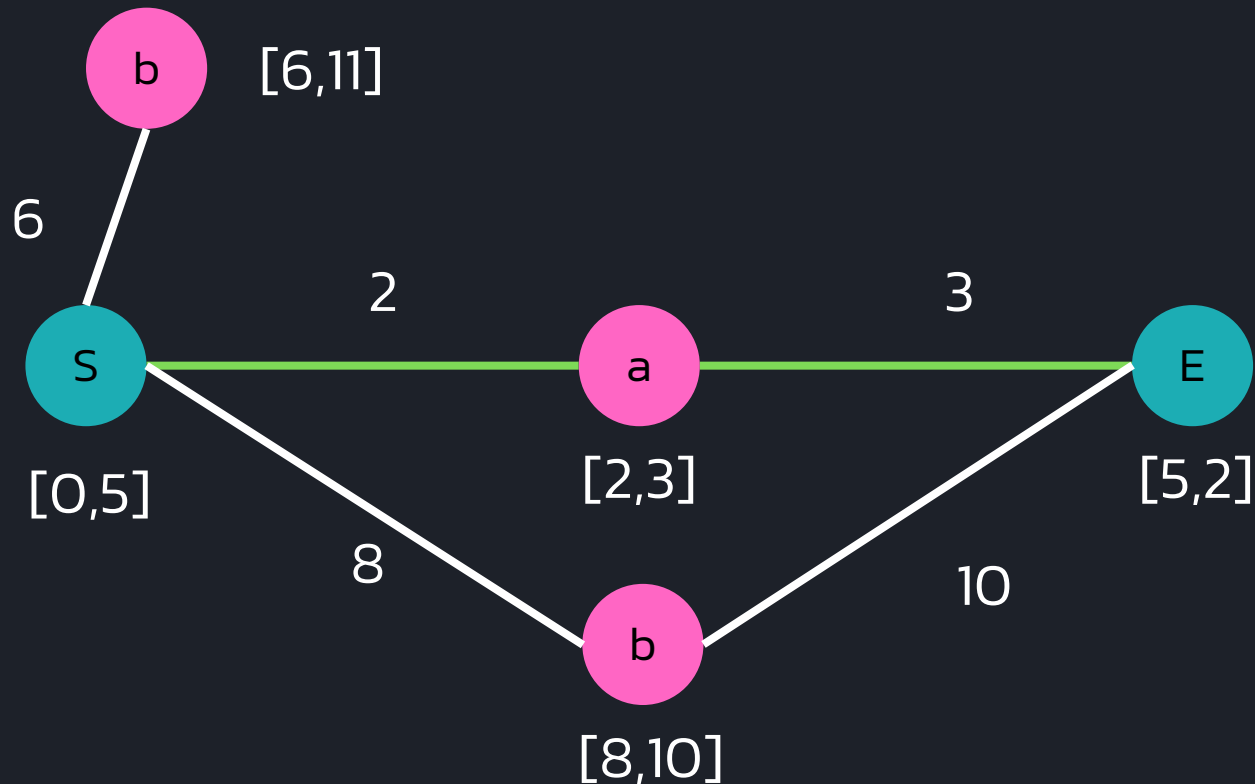


SHORTEST PATH נגזרת NODE ורא

หา NODE ระหว่าง SHORTEST PATH



ถ้าเรา DFS สองรอบจากทั้ง S และ E ระยะทางทั้งสองค่าจะบวกกันได้ระยะทางจาก S→E พอดี





Zombie Land [30 min]

เฉลย ZOMBIE LAND



task ของข้อนี้มี 2 อย่าง

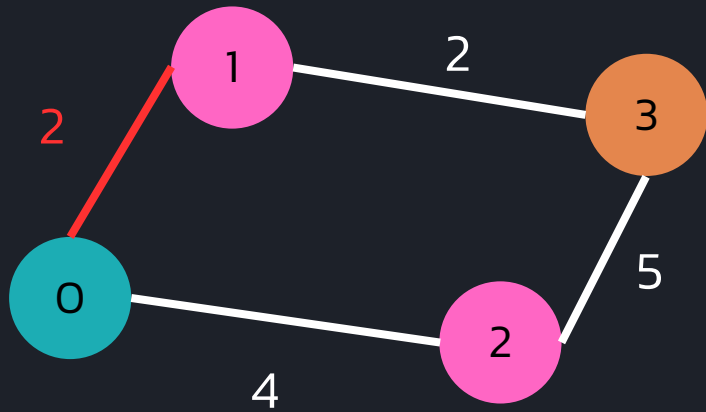
- 1.หาตึกอันตราย โดยตึกอันตรายคือ "ตึกที่อยู่ระหว่างเส้นทางจาก S->E ที่สั้นที่สุด"
- 2.เอาตึกอันตรายทั้งหมดมา DFS ว่าจากตึกพวกนั้น ไปหาตึกอื่นๆได้เร็วที่สุดเท่าไร

DFS $O(N)$ -> สามรอบก็ $O(3N) = O(3*3*1000) = O(9000)$ กันๆเลย



God of War [30 min]

เกม GOD OF WAR

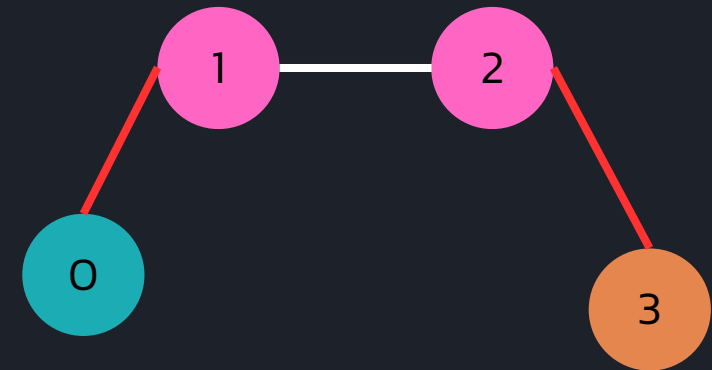


ทำลายตามลำดับ ดังนี้ $5 \rightarrow 4 \rightarrow 2 \rightarrow 2$
พอ $2(1,3)$ ถูกทำลาย ก็หมดทางจาก $S \rightarrow E$ เกมจะจบ

point : หา path ระหว่าง node u และ node v (อาจเป็น S, E เลยก็ได้)
ที่ u ถูกเดินทางมาจาก S ได้และ v ถูกเดินทางมาจาก E ได้
ที่รางวัลของ path ว่าจะมาถึง u และ v มีค่าน้อยสุด

Task 1 : DFS จากจุด S และ E ตามลำดับ หาทางมา u, v ที่น้อยสุด

Task 2 : วนหา node u และ node v ตั้งแต่ $[0, N-1]$ เพื่อดู node u, v



สีแดงรวมกันได้น้อยๆ

HOMework

- น้ํายาเพิ่มความเร็ว
- Royal Parade

