

경제학적 관점으로 분석하는

# 헝가리안 알고리즘

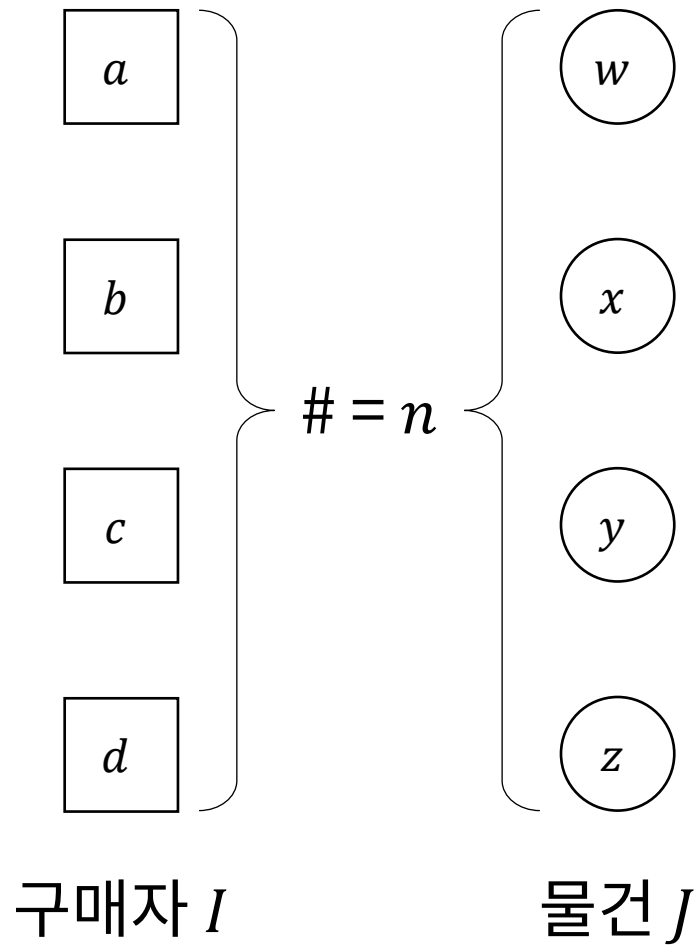
신용호 (연세대학교)

연세 이론 컴퓨터과학 학생 모임 주니어 세미나  
2023. 3. 27.



연세대학교 컴퓨터과학과  
YONSEI UNIVERSITY  
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

# 할당 문제 Assignment problem



$I \backslash J$	$w$	$x$	$y$	$z$	
$a$					
$b$					
$c$					
$d$					

# 할당 문제 Assignment problem

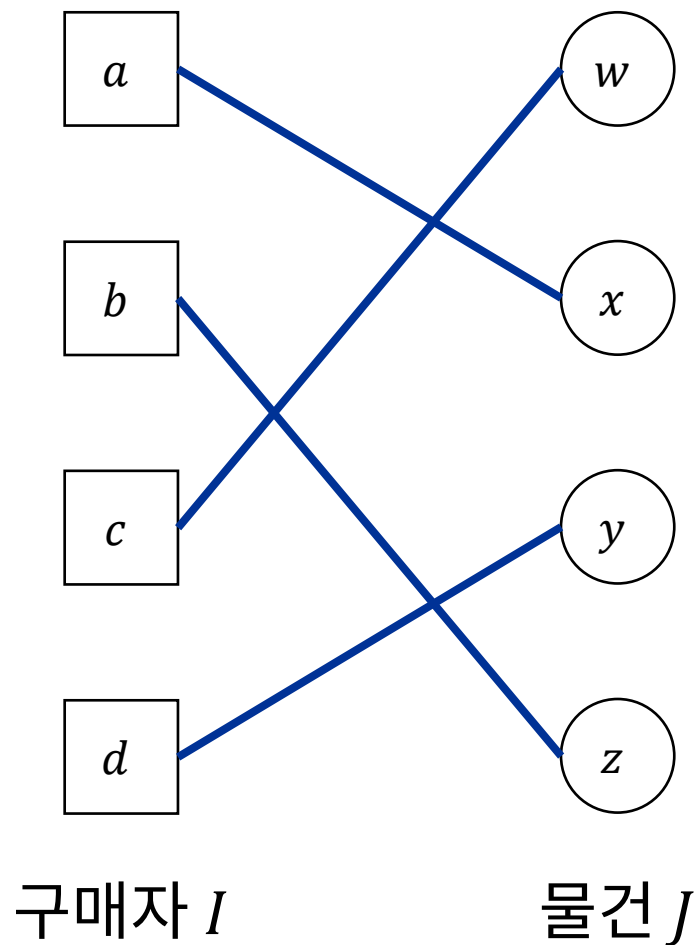


구매자  $I$

물건  $J$

$I \backslash J$	$w$	$x$	$y$	$z$	
$a$					
$b$					
$c$					
$d$					

# 할당 문제 Assignment problem



$I \backslash J$	$w$	$x$	$y$	$z$	
$a$					
$b$					
$c$					
$d$					

# 할당 문제 Assignment problem

$a$

$w$

$b$

$x$

$c$

$y$

$d$

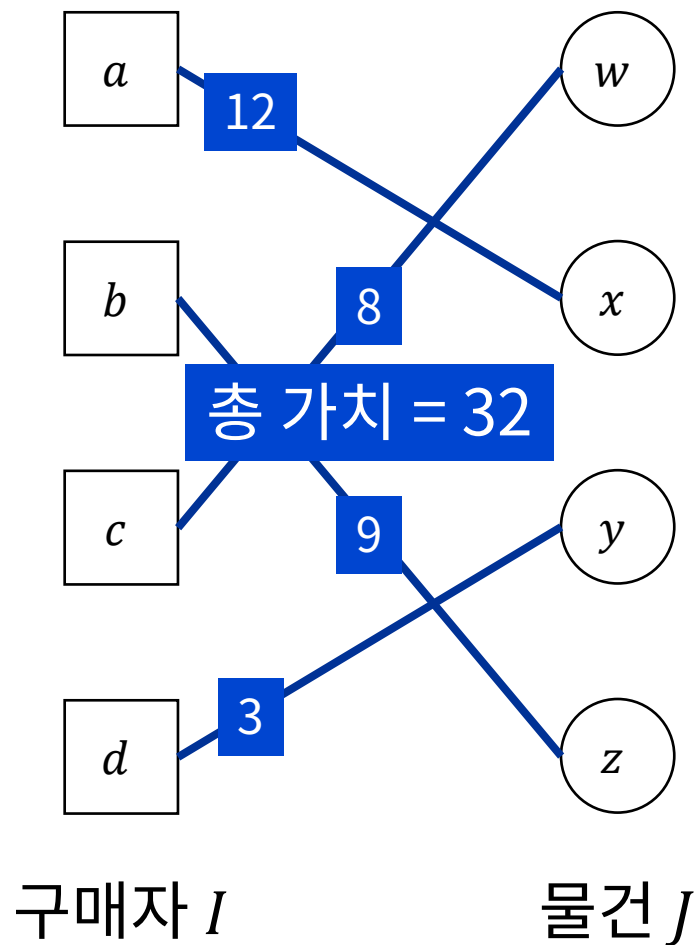
$z$

구매자  $I$

물건  $J$

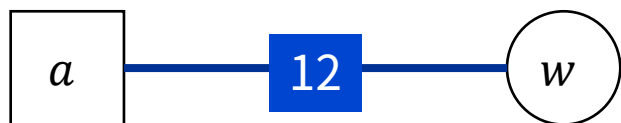
$I \backslash J$	$w$	$x$	$y$	$z$	
$a$	12	12	12	8	
$b$	5	6	10	9	
$c$	8	5	11	11	
$d$	2	3	3	7	

# 할당 문제 Assignment problem

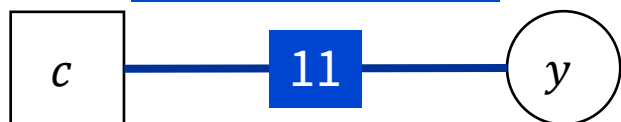


$I \backslash J$	$w$	$x$	$y$	$z$	
$a$	12	12	12	8	
$b$	5	6	10	9	
$c$	8	5	11	11	
$d$	2	3	3	7	

# 할당 문제 Assignment problem



총 가치 = 36



구매자  $I$

물건  $J$

$I \backslash J$	$w$	$x$	$y$	$z$	
$a$	12	12	12	8	
$b$	5	6	10	9	
$c$	8	5	11	11	
$d$	2	3	3	7	

# 할당 문제 Assignment problem

## 입력

구매자  $I$  & 물건  $J$  ( $|I| = |J| = n$ )

각 구매자  $i \in I$ 가 생각하는 물건  $j \in J$ 의 가치  $v_{i,j}$

## 출력

총 가치가 최대인 구매자-물건 할당 방법, 즉

$$\sum_{i \in I} v_{i, \sigma(i)}$$

가 최대인 일대일 대응  $\sigma: I \rightarrow J$



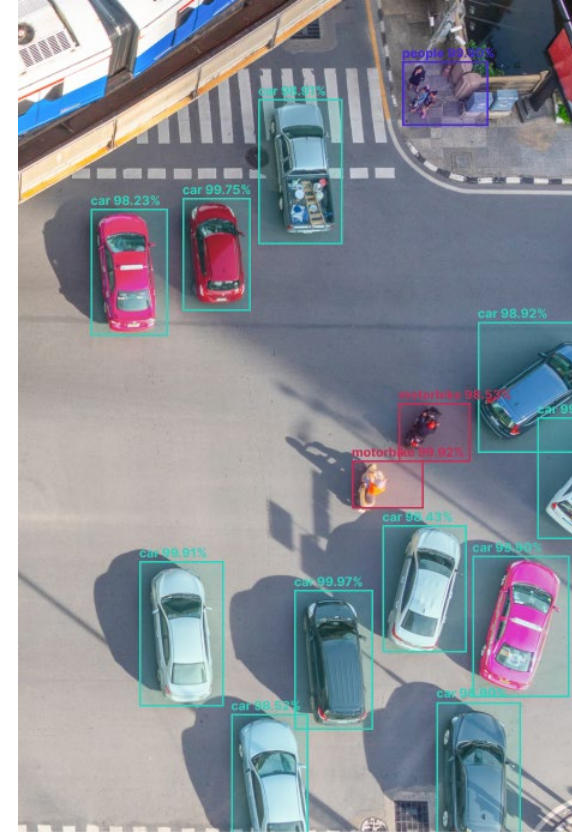
# 활용



택시 배차

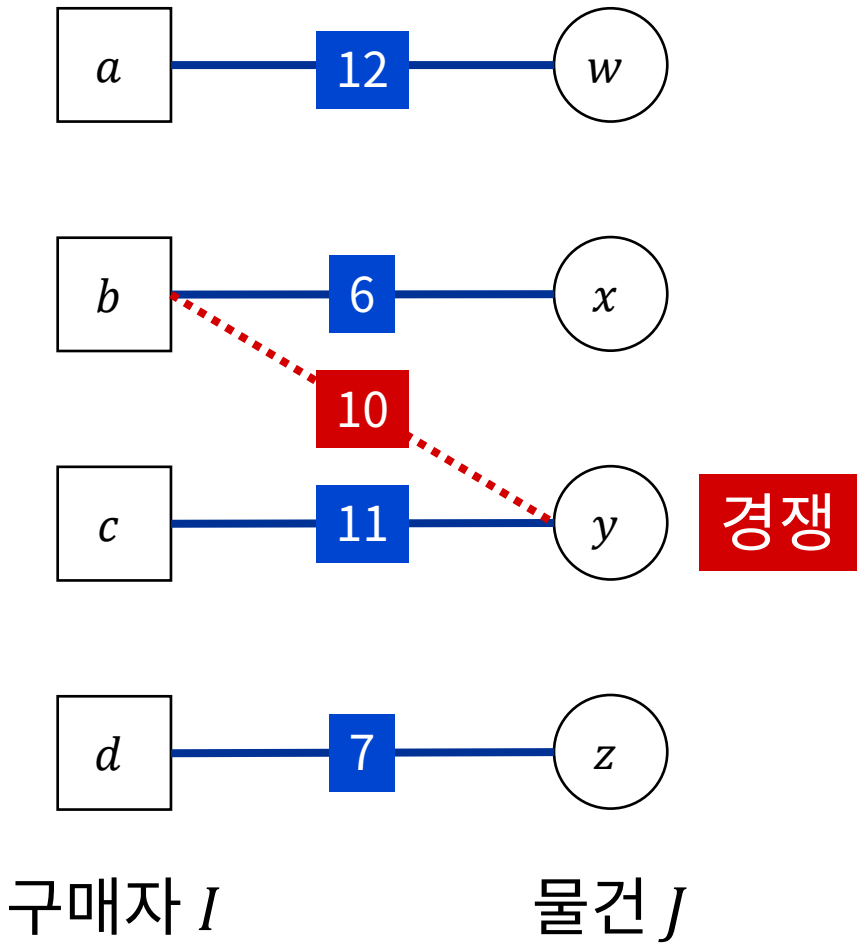


물류 관리



객체 추적

# 깐깐한 구매자



$I \backslash J$	$w$	$x$	$y$	$z$	
$a$	12	12	12	8	
$b$	5	6	10	9	
$c$	8	5	11	11	
$d$	2	3	3	7	

# 경쟁의 해소

## 경쟁

많은 구매자가 적은 수의 물건을 원하는 상황

## 가격

경쟁을 해소하는 경제학적 방법

각 물건  $j \in J$ 마다  $p_j$ 의 가격이 붙었다고 가정

# 경쟁의 해소

## 가격

각 물건  $j \in J$ 마다  $p_j$ 의 가격이 붙었다고 가정

## 효용 Utility

구매자  $i$ 가 물건  $j$ 에 대해 생각하는 이득, 즉,  $v_{i,j} - p_j$

## 균형 Equilibrium

모든 구매자  $i$ 가 자신이 받은 물건  $\sigma(i)$ 를 만족하는 상황, 즉,

$$\forall i \in I \forall j \in J \quad v_{i,\sigma(i)} - p_{\sigma(i)} \geq v_{i,j} - p_j$$

# 목표

균형을 이루면서 총 가치는 최대가 되는 할당과 가격 찾기

# 알고리즘 개요

1. 구매자들끼리 경쟁이 붙은 물건 찾기
2. 경쟁이 붙은 물건의 가격 적당히 인상
3. 경쟁이 없을 때까지 위 작업 반복

# 해결해야 하는 질문

1. 구매자들끼리 경쟁이 붙은 물건 찾기
  2. 경쟁이 붙은 물건의 가격 적당히 인상
  3. 경쟁이 없을 때까지 위 작업 반복
- A. 경쟁이 붙은 물건을 찾는 방법
  - B. 경쟁 중인 물건의 가격 인상폭
  - C. 언젠간 균형에 도달하는지
  - D. 균형 도달 시 총 가치가 최대인지

# 해결해야 하는 질문

1. 구매자들끼리 경쟁이 붙은 물건 찾기
2. 경쟁이 붙은 물건의 가격 적당히 인상
3. 경쟁이 없을 때까지 위 작업 반복

- A. 경쟁이 붙은 물건을 찾는 방법
- B. 경쟁 중인 물건의 가격 인상폭
- C. 언젠간 균형에 도달하는지
- D. 균형 도달 시 총 가치가 최대인지





# 경쟁이 붙은 물건 찾기


$I \backslash J$	$w$	$x$	$y$	$z$	$\max u$
$a$	12	12	12	8	12
$b$	5	6	10	9	10
$c$	8	5	11	11	11
$d$	2	3	3	7	7
$p$	0	0	0	0	

# 경쟁이 붙은 물건 찾기


$I \backslash J$	$w$	$x$	$y$	$z$	$\max u$
$a$	0	0	0	-4	12
$b$	-5	-4	0	-1	10
$c$	-3	-6	0	0	11
$d$	-5	-4	-4	0	7
$p$	0	0	0	0	



# 경쟁이 붙은 물건 찾기

$J'$         

$I'$	$I \backslash J$	$w$	$x$	$y$	$z$	$\max u$
	$a$	0	0	0	-4	12
	$b$	-5	-4	0	-1	10
	$c$	-3	-6	0	0	11
	$d$	-5	-4	-4	0	7
	$p$	0	0	0	0	

# 경쟁이 붙은 물건 찾기

$I'$  

$J'$   

$I \backslash J$	$w$	$x$	$y$	$z$	$\max u$
$a$	0	0	0	-4	12
$b$	-5	-4	0	-1	10
$c$	-3	-6	0	0	11
$d$	-5	-4	-4	0	7
$p$	0	0	0	0	

만약  $|I'| + |J'| < n$  이라면?

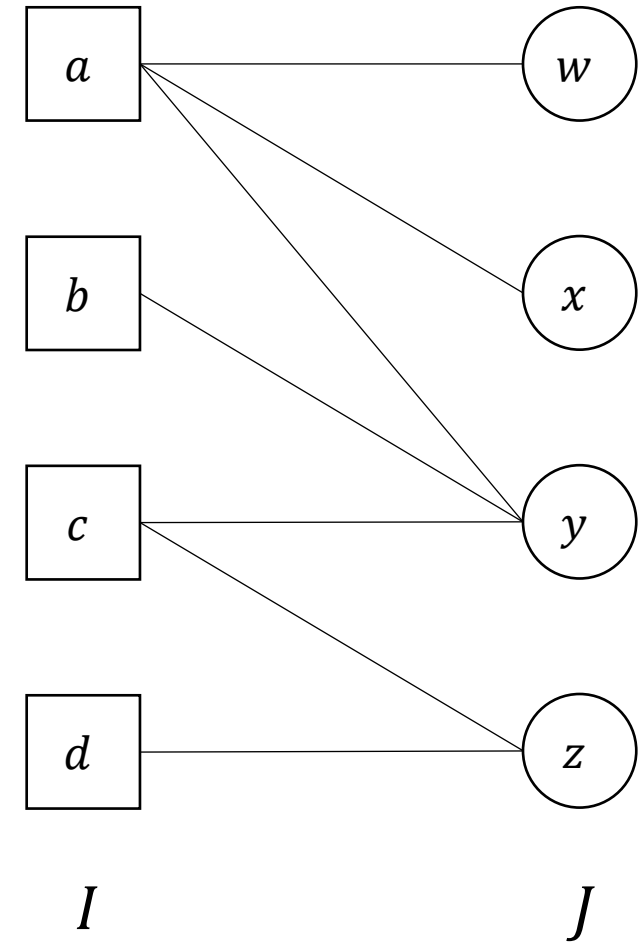
$I \setminus I'$ 은  $J'$ 의 물건만 원함

$$|I \setminus I'| = n - |I'| > |J'|$$


**경쟁 발생!**

# 경쟁이 붙은 물건 찾기

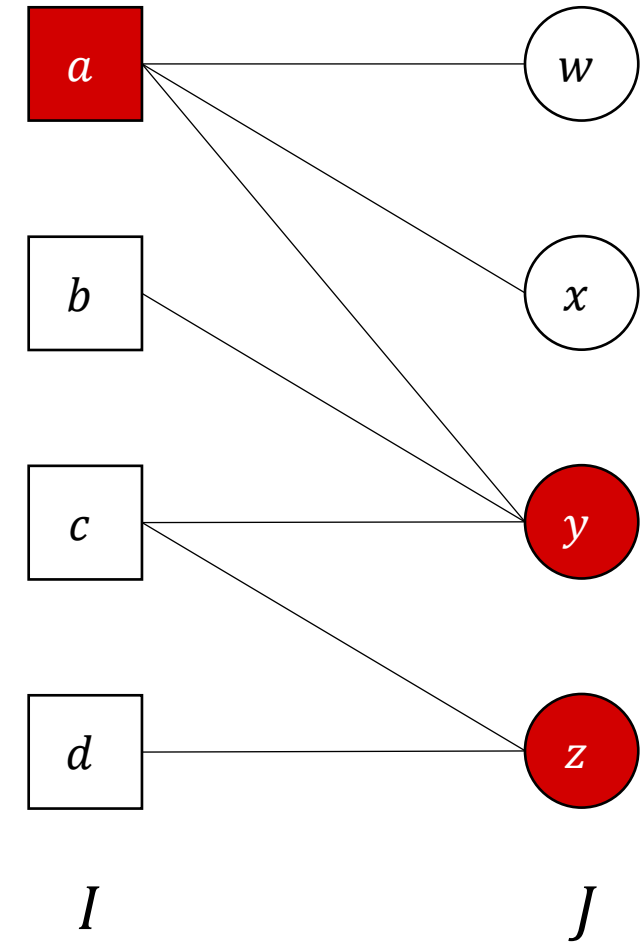
$I \backslash J$	$w$	$x$	$y$	$z$	$\max u$
$a$	0	0	0	-4	12
$b$	-5	-4	0	-1	10
$c$	-3	-6	0	0	11
$d$	-5	-4	-4	0	7
$p$	0	0	0	0	



# 경쟁이 붙은 물건 찾기



$I \backslash J$	$w$	$x$	$y$	$z$	$\max u$
$a$	0	0	0	-4	12
$b$	-5	-4	0	-1	10
$c$	-3	-6	0	0	11
$d$	-5	-4	-4	0	7
$p$	0	0	0	0	



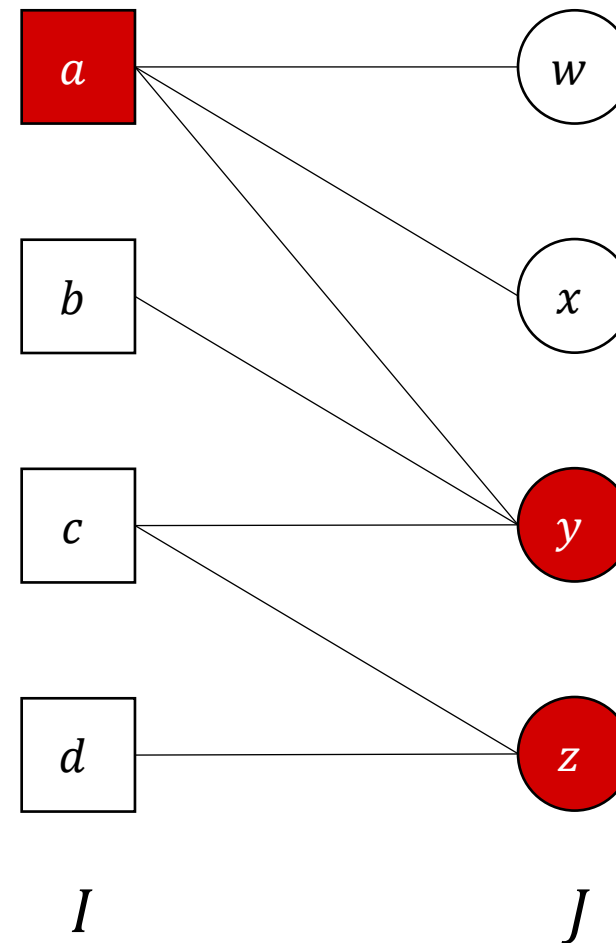
# 경쟁이 붙은 물건 찾기

## 정점 덮개 Vertex cover

어떤 그래프에서 모든 간선의 적어도 한 끝점이 속해 있는 정점의 부분 집합

## 코니그의 정리 König's theorem

이분 그래프에서 최소 정점 덮개의 크기는 최대 이분 매칭의 크기와 동일하다.



# 해결해야 하는 질문

1. 구매자들끼리 경쟁이 붙은 물건 찾기
2. 경쟁이 붙은 물건의 가격 적당히 인상
3. 경쟁이 없을 때까지 위 작업 반복

- A. 경쟁이 붙은 물건을 찾는 방법  
> 정점 뒤통개 & 퀵니그의 정리
- B. 경쟁 중인 물건의 가격 인상폭
- C. 언젠간 균형에 도달하는지
- D. 균형 도달 시 총 가치가 최대인지



# 해결해야 하는 질문

1. 구매자들끼리 경쟁이 붙은 물건 찾기
2. 경쟁이 붙은 물건의 가격 적당히 인상
3. 경쟁이 없을 때까지 위 작업 반복

A. 경쟁이 붙은 물건을 찾는 방법  
> 정점 덮개 & 쿼니그의 정리

**B. 경쟁 중인 물건의 가격 인상폭**

C. 언젠간 균형에 도달하는지

D. 균형 도달 시 총 가치가 최대인지

# 경쟁 중인 물건 가격 인상

$I \backslash J$	$w$	$x$	$y$	$z$	$\max u$
$a$	12	12	12	8	12
$b$	5	6	10	9	10
$c$	8	5	11	11	11
$d$	2	3	3	7	7
$p$	0	0	0	0	

$I \backslash J$	$w$	$x$	$y$	$z$	$\max u$
$a$	0	0	0	-4	12
$b$	-5	-4	0	-1	10
$c$	-3	-6	0	0	11
$d$	-5	-4	-4	0	7
$p$	0	0	0	0	

# 경쟁 중인 물건 가격 인상

$I \backslash J$	$w$	$x$	$y$	$z$	$\max u$
$a$	12	12	12	8	12
$b$	5	6	10	9	7
$c$	8	5	11	11	8
$d$	2	3	3	7	4
$p$	0	0	3	3	

$I \backslash J$	$w$	$x$	$y$	$z$	$\max u$
$a$	0	0	-3	-7	12
$b$	-2	-1	0	-1	7
$c$	0	-3	0	0	8
$d$	-2	-1	-4	0	4
$p$	0	0	3	3	

# 경쟁 중인 물건 가격 인상

$I \backslash J$	$w$	$x$	$y$	$z$	$\max u$
$a$	0	0	0	-4	12
$b$	-5	-4	0	-1	10
$c$	-3	-6	0	0	11
$d$	-5	-4	-4	0	7
$p$	0	0	0	0	

$I \backslash J$	$w$	$x$	$y$	$z$	$\max u$
$a$	0	0	-3	-7	12
$b$	-2	-1	0	-1	7
$c$	0	-3	0	0	8
$d$	-2	-1	-4	0	4
$p$	0	0	3	3	

# 해결해야 하는 질문

1. 구매자들끼리 경쟁이 붙은 물건 찾기
2. 경쟁이 붙은 물건의 가격 적당히 인상
3. 경쟁이 없을 때까지 위 작업 반복

- A. 경쟁이 붙은 물건을 찾는 방법  
> 정점 덮개 & 쿨니그의 정리
- B. 경쟁 중인 물건의 가격 인상폭  
>  $I \setminus I'$ 과  $J \setminus J'$ 의 최소 절댓값
- C. 언젠간 균형에 도달하는지
- D. 균형 도달 시 총 가치가 최대인지

# 해결해야 하는 질문

1. 구매자들끼리 경쟁이 붙은 물건 찾기
2. 경쟁이 붙은 물건의 가격 적당히 인상
3. 경쟁이 없을 때까지 위 작업 반복

A. 경쟁이 붙은 물건을 찾는 방법  
> 정점 덮개 & 쿨니그의 정리

B. 경쟁 중인 물건의 가격 인상폭  
>  $I \setminus I'$ 과  $J \setminus J'$ 의 최소 절댓값

**C. 언젠간 균형에 도달하는지**

D. 균형 도달 시 총 가치가 최대인지

# 알고리즘의 종료

$I \backslash J$	$w$	$x$	$y$	$z$	$\max u$
$a$	0	0	0	-4	12
$b$	-5	-4	0	-1	10
$c$	-3	-6	0	0	11
$d$	-5	-4	-4	0	7
$p$	0	0	0	0	

# 알고리즘의 종료

## 알 수 있는 사실 1

노란 행렬의 원소는 항상 0보다 작거나 같다.



# 알고리즘의 종료

$I \backslash J$	$w$	$x$	$y$	$z$	$\max u$
$a$	0	0	0	-4	12
$b$	-5	-4	0	-1	10
$c$	-3	-6	0	0	11
$d$	-5	-4	-4	0	7
$p$	0	0	0	0	

$I \backslash J$	$w$	$x$	$y$	$z$	$\max u$
$a$	0	0	-3	-7	12
$b$	-2	-1	0	-1	7
$c$	0	-3	0	0	8
$d$	-2	-1	-4	0	4
$p$	0	0	3	3	

# 알고리즘의 종료

$I \backslash J$	$w$	$x$	$y$	$z$	$\max u$
$a$	0	0	0	-4	12
$b$	-5	-4	0	-1	10
$c$	-3	-6	0	0	11
$d$	-5	-4	-4	0	7
$p$	0	0	0	0	

$I \backslash J$	$w$	$x$	$y$	$z$	$\max u$
$a$	0	0	-3	-7	12
$b$	-2	-1	0	-1	7
$c$	0	-3	0	0	8
$d$	-2	-1	-4	0	4
$p$	0	0	3	3	

# 알고리즘의 종료

$I \backslash J$	$w$	$x$	$y$	$z$	$\max u$
$a$	0	0	0	-4	12
$b$	-5	-4	0	-1	10
$c$	-3	-6	0	0	11
$d$	-5	-4	-4	0	7
$p$	0	0	0	0	

$I \backslash J$	$w$	$x$	$y$	$z$	$\max u$
$a$	0	0	-3	-7	12
$b$	-2	-1	0	-1	7
$c$	0	-3	0	0	8
$d$	-2	-1	-4	0	4
$p$	0	0	3	3	

# 알고리즘의 종료

$I \backslash J$	$w$	$x$	$y$	$z$	$\max u$
$a$	0	0	0	-4	12
$b$	-5	-4	0	-1	10
$c$	-3	-6	0	0	11
$d$	-5	-4	-4	0	7
$p$	0	0	0	0	

$I \backslash J$	$w$	$x$	$y$	$z$	$\max u$
$a$	0	0	-3	-7	12
$b$	-2	-1	0	-1	7
$c$	0	-3	0	0	8
$d$	-2	-1	-4	0	4
$p$	0	0	3	3	

# 알고리즘의 종료

$I \backslash J$	$w$	$x$	$y$	$z$	$\max u$
$a$	$I', J \setminus J' \ 0$	$0 \ I', J' \ -4$			12
$b$	-5	-4	0	-1	10
$c$	$I \setminus I', J \setminus J' \ -3$	$I \setminus I', J' \ -5$	$I \setminus I', J' \ 0$		11
$d$	-5	-4	-4	0	7
$p$	0	0	0	0	

$I \backslash J$	$w$	$x$	$y$	$z$	$\max u$
$a$	$I', J \setminus J' \ 0$	$-3 \ I', J' \ -7$			12
$b$	-2	-1	0	-1	7
$c$	$I \setminus I', J \setminus J' \ -3$	$I \setminus I', J' \ -5$	$I \setminus I', J' \ 0$		8
$d$	-2	-1	-4	0	4
$p$	0	0	3	3	

# 알고리즘의 종료

단계 진입 조건

$$|I'| + |J'| < n$$

단계 마다의 증가량

$$\begin{aligned} & \varepsilon \cdot |I \setminus I'| \cdot |J \setminus J'| - \varepsilon \cdot |I'| \cdot |J'| \\ &= \varepsilon \cdot ((n - |I'|)(n - |J'|) - |I'| |J'|) \\ &= \varepsilon \cdot (n^2 - (|I'| + |J'|)n) \\ &> 0 \end{aligned}$$

# 알고리즘의 종료

## 알 수 있는 사실 1.

노란 행렬의 원소는 항상 0보다 작거나 같다.

## 알 수 있는 사실 2.

단계를 진행할 때마다 노란 행렬의 원소의 합은 커지기만 한다.

# 해결해야 하는 질문

1. 구매자들끼리 경쟁이 붙은 물건 찾기
2. 경쟁이 붙은 물건의 가격 적당히 인상
3. 경쟁이 없을 때까지 위 작업 반복

- A. 경쟁이 붙은 물건을 찾는 방법  
> 정점 덮개 & 쿨니그의 정리
- B. 경쟁 중인 물건의 가격 인상폭  
>  $I \setminus I'$ 과  $J \setminus J'$ 의 최소 절댓값
- C. 언젠간 균형에 도달하는지  
> 노란 행렬 원소 합 증가
- D. 균형 도달 시 총 가치가 최대인지



# 해결해야 하는 질문

1. 구매자들끼리 경쟁이 붙은 물건 찾기
2. 경쟁이 붙은 물건의 가격 적당히 인상
3. 경쟁이 없을 때까지 위 작업 반복

A. 경쟁이 붙은 물건을 찾는 방법  
> 정점 덮개 & 쿨니그의 정리

B. 경쟁 중인 물건의 가격 인상폭  
>  $I \setminus I'$ 과  $J \setminus J'$ 의 최소 절댓값

C. 언젠간 균형에 도달하는지  
> 노란 행렬 원소 합 증가

**D. 균형 도달 시 총 가치가 최대인지**

# 최대 총 가치 할당

$I \backslash J$	$w$	$x$	$y$	$z$	$\max u$
$a$	12	12	12	8	12
$b$	5	6	10	9	7
$c$	8	5	11	11	8
$d$	2	3	3	7	4
$p$	0	0	3	3	

총 가치

$$12+10+8+7 \\ = 37$$

최대 효용 + 가격

$$12+7+8+4+3+3+0+0 \\ = 37$$

# 최대 총 가치 할당

## 정리 1.

임의의 가격  $p \in \mathbb{R}^J$ , 임의의 할당  $\sigma: I \rightarrow J$ 에 대해

$$\sum_{i \in I} \max u_i + \sum_{j \in J} p_j \geq \sum_{i \in I} v_{i, \sigma(i)}$$

[증명]

$$\begin{aligned} \sum_{i \in I} \max u_i + \sum_{j \in J} p_j &= \sum_{i \in I} \max_{j \in J} (v_{i,j} - p_j) + \sum_{j \in J} p_j \\ &\geq \sum_{i \in I} (v_{i, \sigma(i)} - p_{\sigma(i)}) + \sum_{j \in J} p_j \\ &= \sum_{i \in I} v_{i, \sigma(i)} \end{aligned}$$

# 최대 총 가치 할당

## 정리 2.

어떤 가격  $p \in \mathbb{R}^J$ 에서 균형에 도달한 할당  $\sigma: I \rightarrow J$ 에 대해

$$\sum_{i \in I} \max u_i + \sum_{j \in J} p_j = \sum_{i \in I} v_{i, \sigma(i)}$$

[증명]

$$\begin{aligned} \sum_{i \in I} \max u_i + \sum_{j \in J} p_j &= \sum_{i \in I} \max_{j \in J} (v_{i,j} - p_j) + \sum_{j \in J} p_j \\ &= \sum_{i \in I} (v_{i, \sigma(i)} - p_{\sigma(i)}) + \sum_{j \in J} p_j \\ &= \sum_{i \in I} v_{i, \sigma(i)} \end{aligned}$$

# 해결해야 하는 질문

1. 구매자들끼리 경쟁이 붙은 물건 찾기
2. 경쟁이 붙은 물건의 가격 적당히 인상
3. 경쟁이 없을 때까지 위 작업 반복

A. 경쟁이 붙은 물건을 찾는 방법  
> 정점 덮개 & 쿨니그의 정리

B. 경쟁 중인 물건의 가격 인상폭  
>  $I \setminus I'$ 과  $J \setminus J'$ 의 최소 절댓값

C. 언젠간 균형에 도달하는지  
> 노란 행렬 원소 합 증가

D. 균형 도달 시 총 가치가 최대인지  
> 총 가치 = 최대 효용 + 가격

# 헝가리안 알고리즘

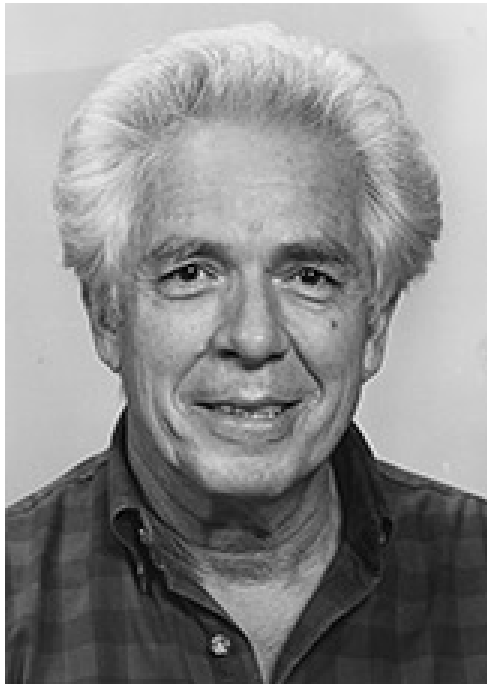
1. 구매자들끼리 경쟁이 붙은 물건 찾기
2. 경쟁이 붙은 물건의 가격 적당히 인상
3. 경쟁이 없을 때까지 위 작업 반복

- A. 경쟁이 붙은 물건을 찾는 방법  
> 정점 덮개 & 쿨니그의 정리
- B. 경쟁 중인 물건의 가격 인상폭  
>  $I \setminus I'$ 과  $J \setminus J'$ 의 최소 절댓값
- C. 언젠간 균형에 도달하는지  
> 노란 행렬 원소 합 증가
- D. 균형 도달 시 총 가치가 최대인지  
> 총 가치 = 최대 효용 + 가격

# 헝가리안 알고리즘

1. 모든 물건  $j \in J$ 에 대해,  $p_j \leftarrow 0$
2. 노란 행렬을 만들고 최소의 개수로 행렬의 0을 모두 덮는 방법  $I' \subseteq I, J' \subseteq J$  찾는다
3. 만약  $|I'| + |J'| < n$  이면,  
 $\varepsilon \leftarrow$  노란 행렬의  $I \setminus I', J \setminus J'$  중 최소 절댓값  
경쟁이 붙은  $j' \in J'$ 에 대해,  $p_{j'} \leftarrow p_{j'} + \varepsilon$
4. 만약  $|I'| + |J'| = n$  이면,  
각 구매자에게 원소 값이 0인 물건을 할당한다.

# 헝가리안 알고리즘



Harold W. Kuhn



Dénes König



Jenő Egerváry



# 생각해 볼 질문

Q1.

최소 비용 할당을 구하는 방법?

Q2.

다항 시간에 계산하는 방법?

Q3.

구매자가 거짓을 고할 수 있다면?

# 감사합니다

질문은 언제나 환영합니다



연세대학교 컴퓨터과학과  
YONSEI UNIVERSITY  
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING