

통계계산특론 I 기말 프로젝트

# Graph Coloring

---

212STG34 강채리

212STG35 노예림

212STG36 백나림

212STG39 유찬미

# CONTENTS

01 서론

02 알고리즘

03 시뮬레이션

04 알고리즘 활용

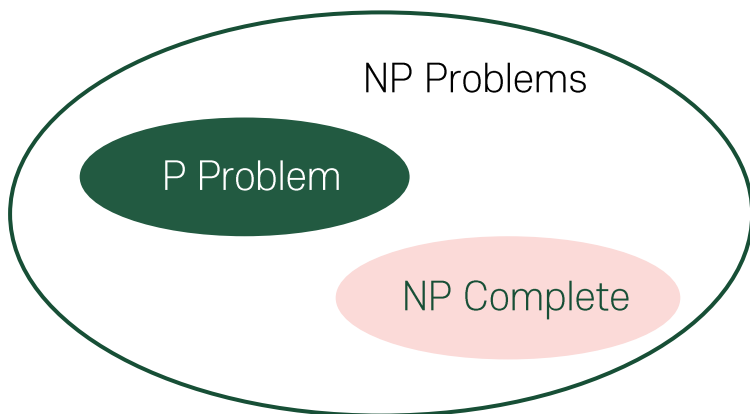
# 01 서론

- NP problem
- Graph Colorig

# 01. 서론

## NP 문제란?

NP 문제란 해를 찾는 방법이 아닌 해를 추측하는 방식



### P 문제

다항 시간(polynomial time)에  
결정론적(deterministically)으로  
해결 가능한 문제들의 집합

### NP 문제

다항 시간(polynomial time)에  
비결정론적(nondeterministically)으로  
해결 가능한 문제들의 집합

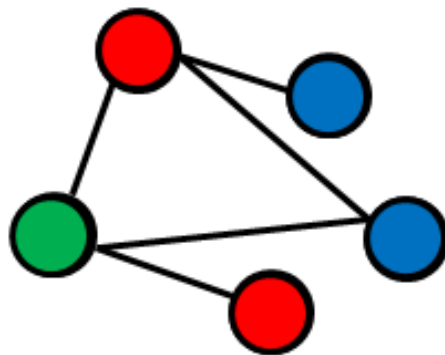
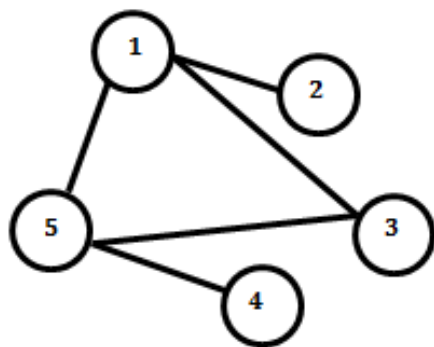
- \* 결정론적 : 인스턴스에 대해 항상 같은 결과가 나오는 것
- \* 비결정론적 : 무엇인가 불확실한 것이 있어 시행하는 방식에 있어 다른 결과가 나올 수 있는 것
- \* NP hard : 다항식 시간 내에 풀 수 없는 문제를 NP hard problem이라고 한다
- \* NP complete : NP hard이면서 동시에 NP 문제인 경우 이를 NP complete problem이라고 한다.

## 01. 서론

### Graph Coloring이란?

- vertex coloring : 인접한 두 노드(vertex)의 색이 겹치지 않게 칠하는 것
- edge coloring : 인접한 두 edge의 색이 겹치지 않게 칠하는 것

**Goal** 최소한의 색깔을 사용하는 것



그래프의 꼭짓점 또는 가장자리 수가 증가함에 따라 문제의 복잡성도 증가

## 01. 서론

### 수학적 정의

A  $k$ -coloring of an undirected graph  $G(V, E)$  is a function

$$\phi : V \rightarrow \{1, \dots, k\}$$

that assign distinct values to adjacent vertices : this is,  $(u, v) \in E \Rightarrow \phi(u) \neq \phi(v)$ .

If  $G$  has a  $k$ -coloring then it is said to be  $k$ -colorable.

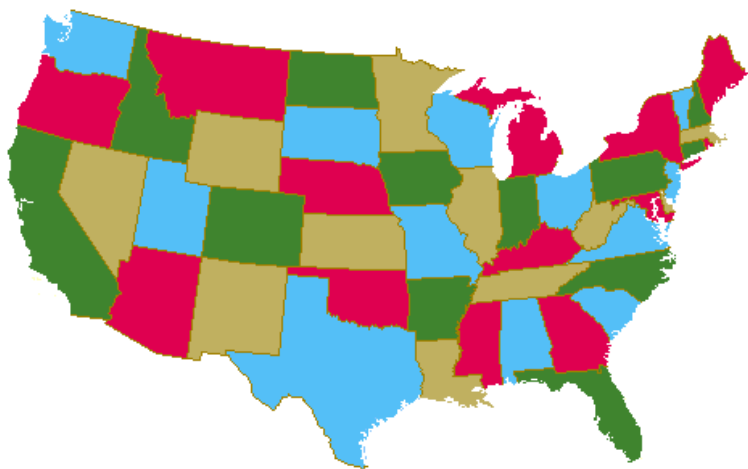
### 용어 정리

- $V$  : vertex(=node, 점)의 개수
- $E$  : edge(연결선)의 개수
- Chromatic Number : 그래프  $G$ 를 채색하는 데 필요한 최소 색깔의 수

## 01. 서론

### The 4-Color Theorem

모든 평면은 4색이면 칠할 수 있다.



#### 문제의 기원

1850년대, Francis Gurthries가  
영국 지도를 색칠하던 중  
서로 다른 4가지의 색으로  
모두 칠할 수 있다는 것을 발견

#### 문제의 증명

- 1870년대 후반 ~ 1880년대  
: 다양한 수학적 증명 방법이 나옴
- 1970년대 : 컴퓨터를 통해 증명  
(1200시간 소요)

## 01. 서론

---

# Graph Coloring의 활용

color은 labeling일 뿐이라는 점을 고려하면 다양한 분야 적용 가능

**01** 스도쿠 퍼즐 풀기

**02** 좌석 배치하기

**03** 택시 배차 배정하기

**04** 시간표 작성하기



# 02 알고리즘

• Greedy • Ant colony optimization • Backtracking

## 02. 알고리즘

### Algorithm

알고리즘에는 크게 두 종류의 알고리즘 존재

알고리즘 종류	정확하게 찾는 알고리즘	근사적으로 찾는 알고리즘
solution	최적인 해 (optimal solution)	적합한 해 (feasible solution)
examples	Backtracking	Greedy, Ant Colony Optimization

- Worst-Case Time Complexity :  $O(n) = 1 + m + m^2 + m^3 + \dots + m^3 = \frac{m^{n+1} - 1}{m - 1} \in \Theta(m^n)$
- Backtracking은 모든 경우의 수를 탐색하므로  $m^n$ 개의 인스턴스를 계산해야 함
- $n$ (노드 수)이 적을 때는 연산이 어렵지 않지만,  $n$ 이 커지면 계산 속도가 기하급수적으로 증가
- 본 프로젝트에서는 Backtracking의 원래 작동방식 알고리즘을 이용하지 않고, 처음으로 해가 나오면 해당 해를 사용하는 알고리즘을 사용
- 대부분의 논문에서도 동일한 방식으로 Backtracking 알고리즘을 GCP에 사용

## 02. 알고리즘

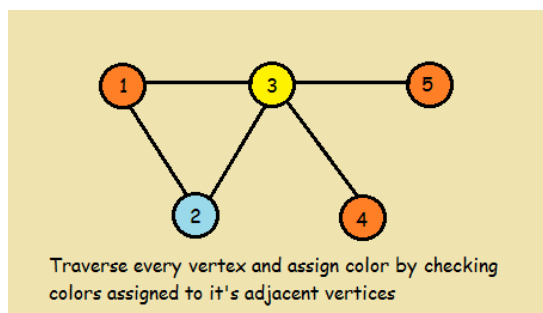
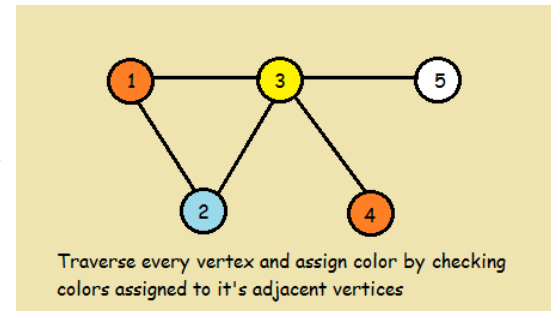
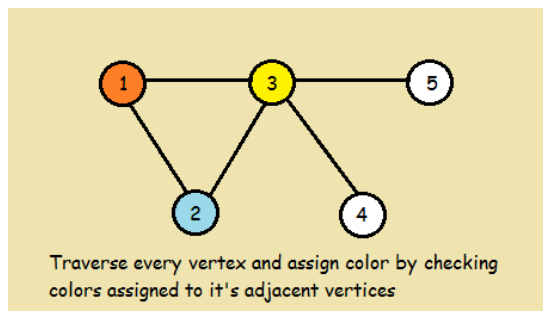
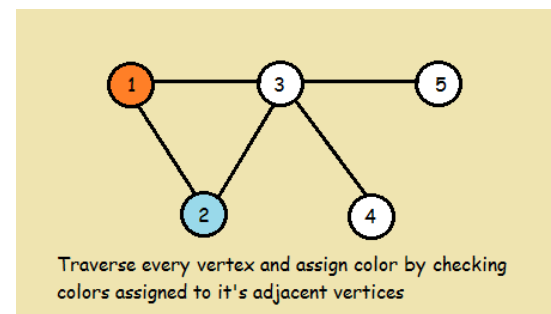
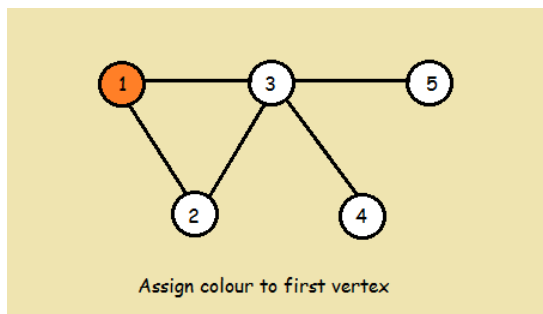
### Greedy

- 최적해를 구할 때 사용하는 **근사적인** 방법
- Greedy는 '탐욕스러운, 욕심 많은'이라는 뜻
- 선택의 순간마다 **당장 눈앞에 보이는 최적**의 상황만을 쫓아 최종적인 해답에 도달하는 방법
- 따라서 로컬 optimum일수는 있으나, 최종적 해답이 아닐 수 있음
- 실용적인 솔루션을 신속하게 생성하지만, 비교적 필요한 색상 수에 따라 '불량'할 수도 있음

## 02. 알고리즘

### Greedy

- 1 노드들에 1부터 마지막까지 번호를 매긴다.
- 2 초기 1번 노드에 대해 랜덤으로 색상을 할당한다.
- 3 번호를 1 증가 시킨 후 해당 번호의 노드에 색을 칠한다.
- 4 이 때, 인접한 노드들과는 다른 색을 칠한다.



## 02. 알고리즘

## Greedy

**Require:**  $S$  = Class Set,  $V$  = Colorless vertices in random order.

**Ensure:**  $S = \emptyset$

**for**  $v$  in  $V$  **do**

**for**  $i$  to  $S.length()$  **do**

**if**  $NonConflitiveEdges(v \cup S_i)$  **then**

$AssignClass(v, S_i)$

            Next Vertex (break)

**end if**

**end for**

**if**  $NotColored(v)$  **then**

$S_{i+1} = NewClass$

$AssignClass(v, S_{i+1})$

        Next Vertex

**end if**

**end for**

- 임의로 생성되는 초기 순열을 사용
- 각 정점에 색상이 지정되어 충돌을 일으키지 않고 색상에 포함될 수 있는지 각 인스턴스에서 비교
- 최악의 경우, 알고리즘 복잡도는  $O(n^2)$

$$P = 0 + 1 + 2 + \dots + n = \frac{n(n+1)}{2}$$

## 02. 알고리즘

### Ant Colony Optimization (ACO)

- 조합 최적화 문제를 해결하기 위한 메타휴리스틱 접근 방식
- 실제 개미의 채집 행동에서 영감을 얻은 진화적 방법으로, 먹이와 둥지 사이의 최단경로를 찾을 수 있음
- 각 단계에서 이전 반복에서 얻은 정보와 각 요소의 적합성을 고려

**1** 개미들은 무작정 개미집 주변을 돌아다닌다.

**2** 만약 음식을 찾아내면 개미는 페로몬을 뿌리며 집으로 돌아온다.

**3** 페로몬은 매우 매력적이어서 개미가 따라가고 싶어 만든다.

**4** 개미가 집으로 돌아오는 횟수가 많을수록 그 경로는 더 견고해진다.

**5** 긴 경로와 짧은 경로가 있으면 같은 시간에 짧은 경로로 이동할 수 있는 횟수가 많다.

**6** 짧은 경로는 갈수록 더 많은 페로몬이 뿌려지면서 더욱 견고해진다.

**7** 페로몬은 휘발성으로 시간이 지나면서 긴 경로는 사라진다.

**8** 결국 모든 개미가 짧은 경로를 선택한다.

## 02. 알고리즘

### Ant Colony Optimization (ACO)

#### pheromone trail evaporation (페로몬 증발)

망각의 유용한 형태이며 새로운 검색 공간 영역의 탐색을 선호

#### daemon actions

개미가 찾은 경로를 관찰하고 최상의 솔루션을 구축한 개미가 사용하는 구성 요소에 여분의 페로몬(추가 페로몬)을 침전

#### TABU : local minimum에 대한 정보 (금지조건)

- 임의의 현재 해에서 이동 가능한 모든 이웃 해를 고려해서 이웃 해 집합 구성
- 이웃 해 집합에서 TABU가 아닌 좋은 해를 선택
- TABU를 계속해서 추가, 제거하는 수정 과정을 거치며 반복
- TABU를 제외한 이웃 해가 없거나, 최적 해를 찾으면 STOP

## 02. 알고리즘

### Ant Colony Optimization (ACO)

*Begin ACS1*

*Initialization; (Pheromone and parameters) While (stop criterion not satisfied) do*

*Position ants on starting vertices Repeat*

*For (each ant) do*

*Choose a vertex to color by applying the transition rule (\*); Tabu list Update;*

*Update Online stage by stage of pheromone trails*

*End for*

*Until (each ant construct a solution)*

*Select the best solution ;*

*Pheromone Offline update of the best solution*

*Evaporation if necessary*

*End While End*

*ACS1*

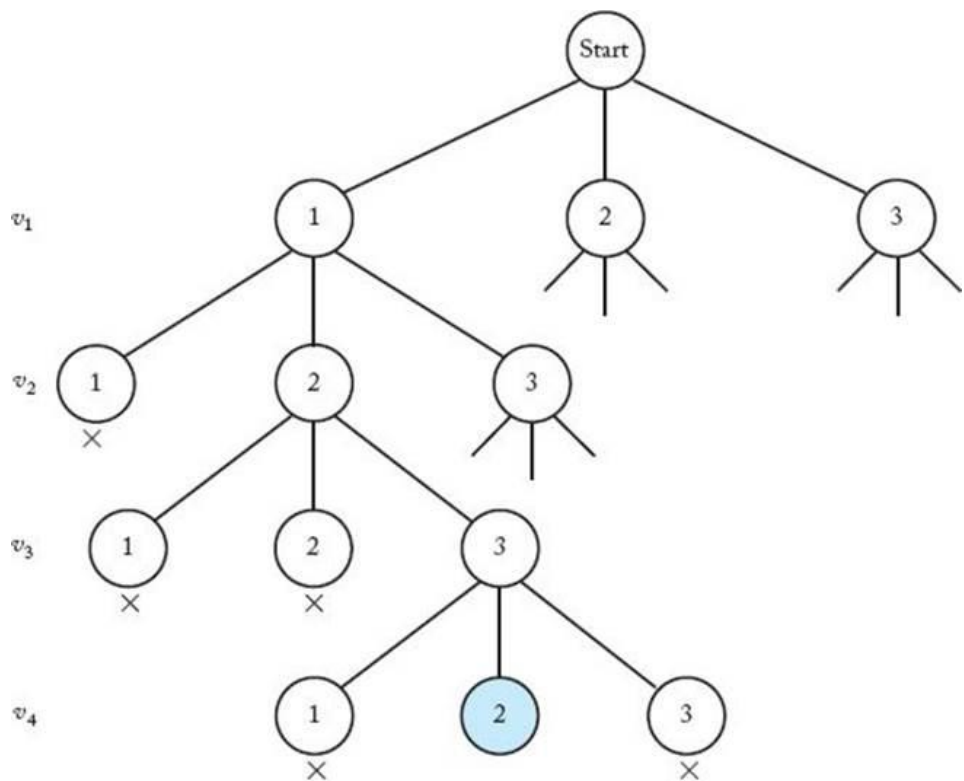
알고리즘 복잡도 :  $O(n^3)$



## 02. 알고리즘

### Backtracking

조건을 만족하는 모든 조합의 수 찾기



- 깊이우선탐색(Depth First Search, DFS)를 사용하여 만약 조건에 맞지 않으면 그 즉시 중단하고 이전으로 돌아가 다시 확인하는 것을 반복하면서 원하는 조건을 찾는 알고리즘
- 가지치기(pruning) : 가능성이 없는 조합은 제거해 나가면서 답을 해결

## 02. 알고리즘

### Backtracking

```
def m_coloring(i):
    global count # 현재 발견한 solution 개수
    global m # 현재 사용 색깔 개수
    # 탐색
    if promising(i):
        if i == n: # 끝까지 탐색 -> 정답
            count+=1
        else:
            for color in range(1, m + 1): # 모든 가능 색깔 탐색
                vcolor[i + 1] = color
                m_coloring(i + 1) # 다음 재귀로 넘어가고 불가능하다면 backTracking

def promising(i):
    j = 1 #SST를 탐색
    flag = True

    while j < i and flag:
        if W[i][j] and vcolor[i] == vcolor[j]: #W[i][j]가 연결되어 있으며, 두 노드의 색깔이 같다면
            flag = False
        j += 1
    return flag
```

알고리즘 복잡도

$$O(n) = 1 + m + m^2 + m^3 + \dots + m^3 = \frac{m^{n+1} - 1}{m - 1} \in \Theta(m^n)$$

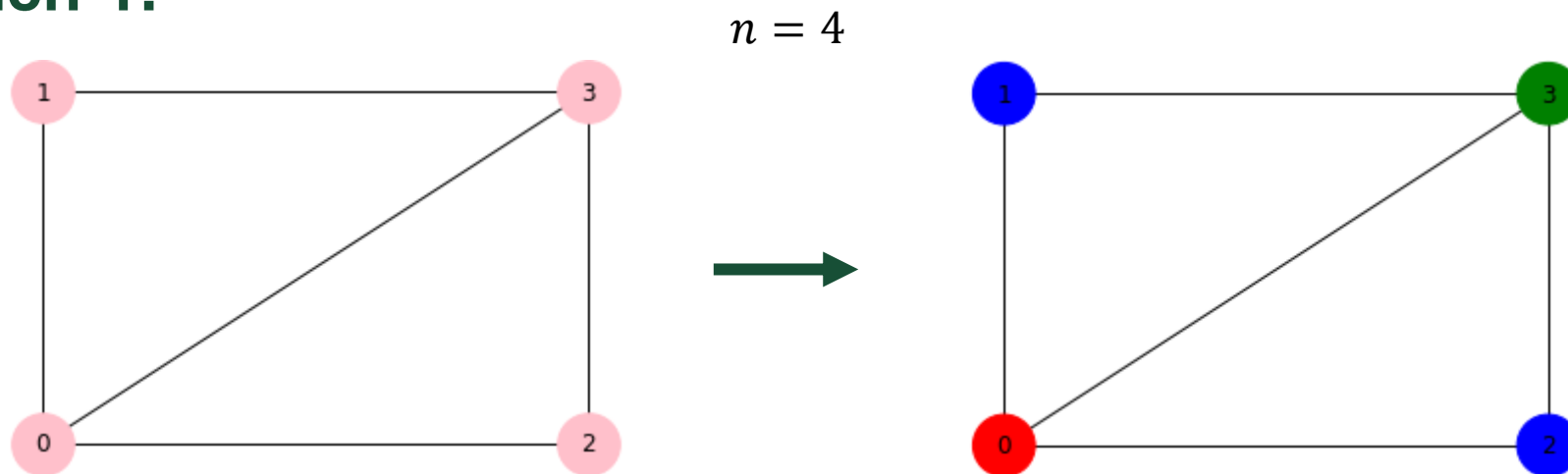
03

## 시뮬레이션

• Simulation • Conclusion

### 03. 시뮬레이션

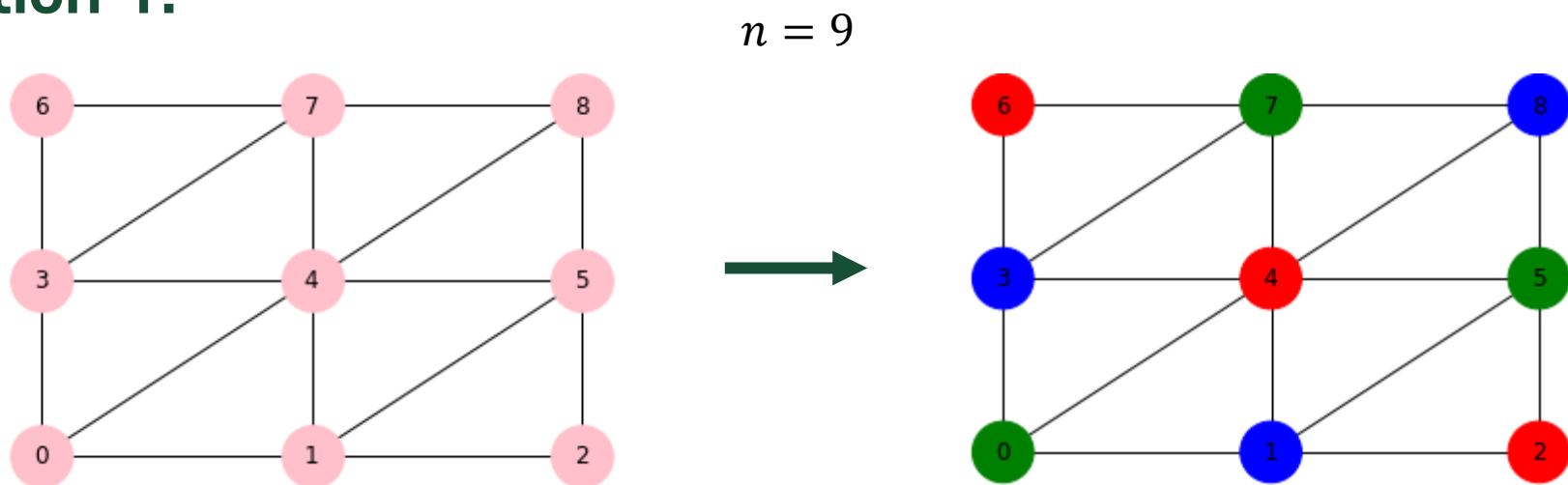
#### Simulation 1.



	Greedy	ACO	Backtracking
Number of Colors	3	3	3
Number of Iterations	4	1	1
Computation time	0.0000 sec	0.0002 sec	0.0008 sec

# 03. 시뮬레이션

## Simulation 1.



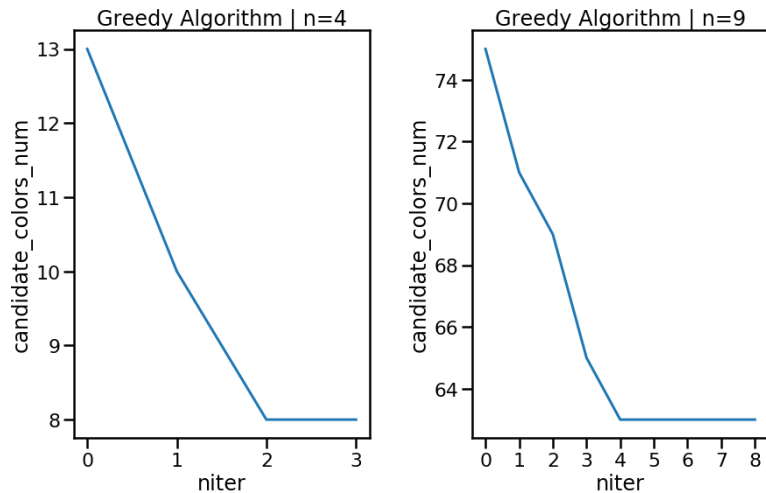
	Greedy	ACO	Backtracking
Number of Colors	3	3	4
Number of Iterations	9	1	1
Computation time	0.0000 sec	0.0303 sec	0.0011 sec

## 03. 시뮬레이션

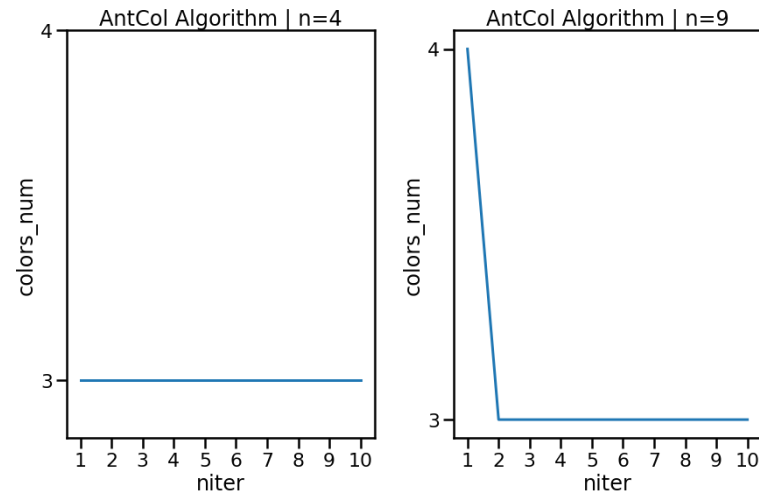
### Simulation 1.

알고리즘 수렴 여부에 대한 비교

Greedy 알고리즘



AntCol 알고리즘



- 그래프의 노드 수(n)가 적을 때는 두 알고리즘 모두 잘 수렴한다.

## 03. 시뮬레이션

### Simulation 2. 그래프 연결성 증가 ( $a$ )

- 무작위로 생성된 그래프를 이용해서 테스트를 수행
- $n$ 은 노드(node) 수,  $a$ 는 연결선 수를 의미
- $n = 100$ 으로 동일한 상황에서  $a$ 를 증가시켜 가면서 각 알고리즘의 성능을 비교

$$n = 100, a = 1638$$

	Greedy	ACO	Backtracking
Number of Colors	15	14	16
Computation time	0.0118 sec	12.4948 sec	0.0229 sec

## 03. 시뮬레이션

Simulation 2. 그래프 연결성 증가 ( $a$ ) $a = 2467$ 

	Greedy	ACO	Backtracking
Number of Colors	19	18	21
Computation time	0.0128 sec	13.3168 sec	0.0140 sec

 $a = 3747$ 

	Greedy	ACO	Backtracking
Number of Colors	33	29	34
Computation time	0.0146 sec	14.7023 sec	0.0275 sec

 $a = 4124$ 

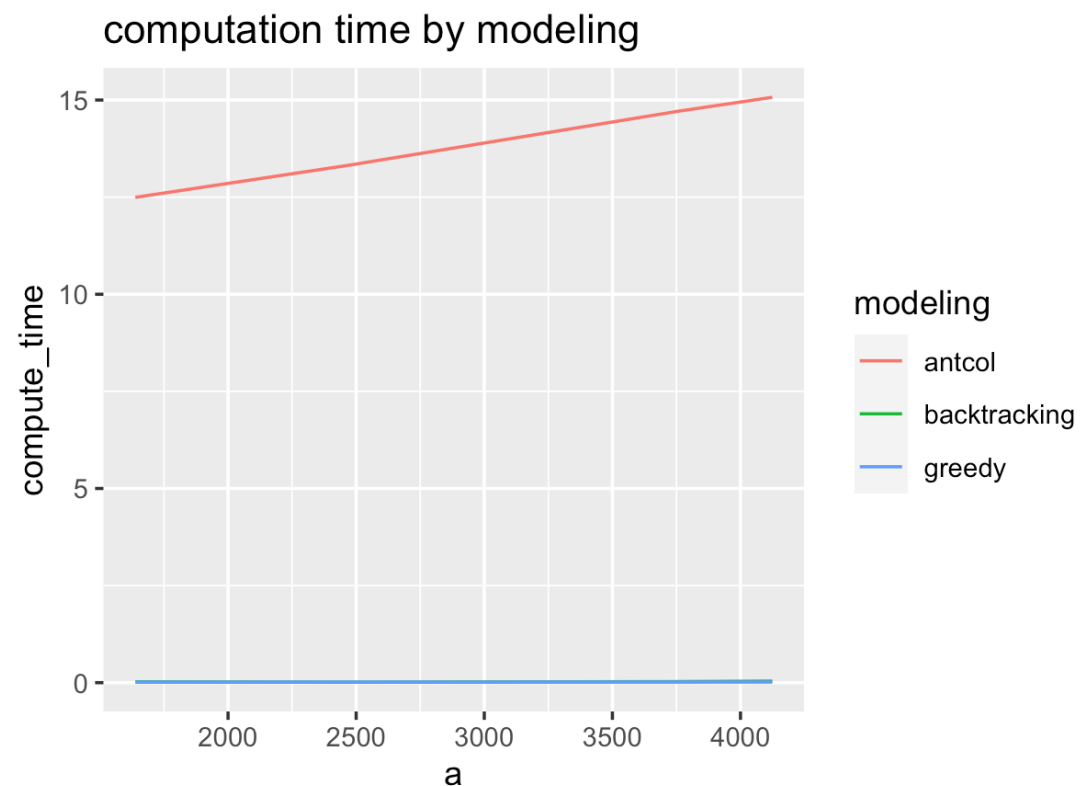
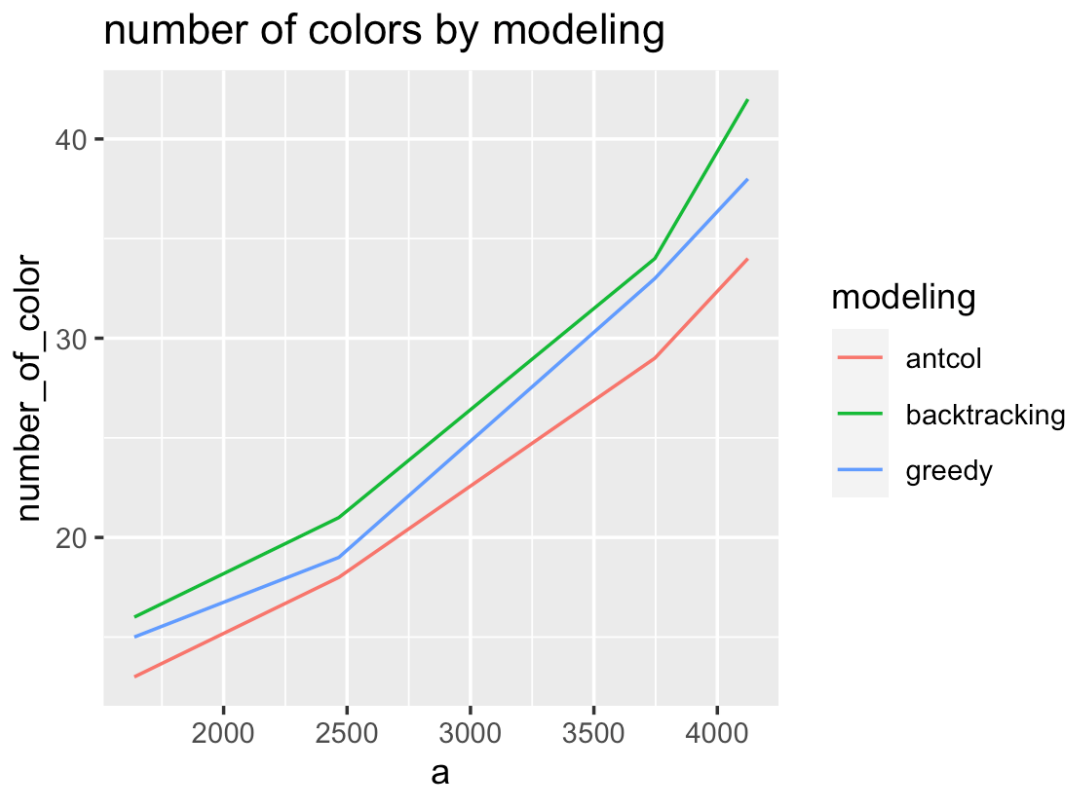
	Greedy	ACO	Backtracking
Number of Colors	38	34	42
Computation time	0.0158 sec	15.0701 sec	0.0402 sec



## 03. 시뮬레이션

Simulation 2. 그래프 연결성 증가 ( $a$ )

연결성 증가에 따른 비교 분석



### 03. 시뮬레이션

#### Simulation 3. 그래프 크기의 증가 ( $n, a$ )

- 무작위로 생성된 그래프를 이용해서 테스트를 수행
- $n$ 은 노드(node) 수,  $a$ 는 연결선 수를 의미

$$n = 100, a = 1638$$

	Greedy	ACO	Backtracking
Number of Colors	15	14	16
Computation time	0.0118 sec	12.4948 sec	0.0229 sec

## 03. 시뮬레이션

Simulation 3. 그래프 크기의 증가 ( $n, a$ ) $n = 500, a = 41959$ 

	Greedy	ACO	Backtracking
Number of Colors	48	43	49
Computation time	0.8166 sec	1456.0349 sec	0.4818 sec

 $n = 800, a = 3747$ 

	Greedy	ACO	Backtracking
Number of Colors	68	63	69
Computation time	3.1208 sec	6214.8627 sec	1.5928 sec

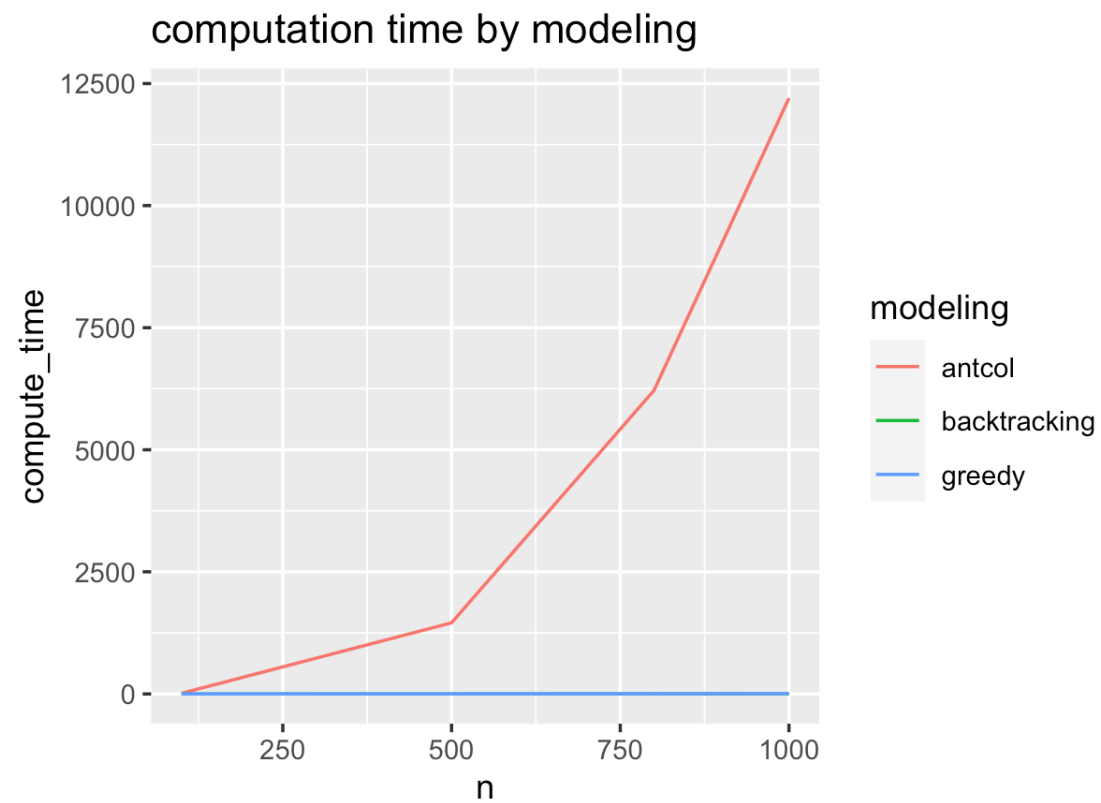
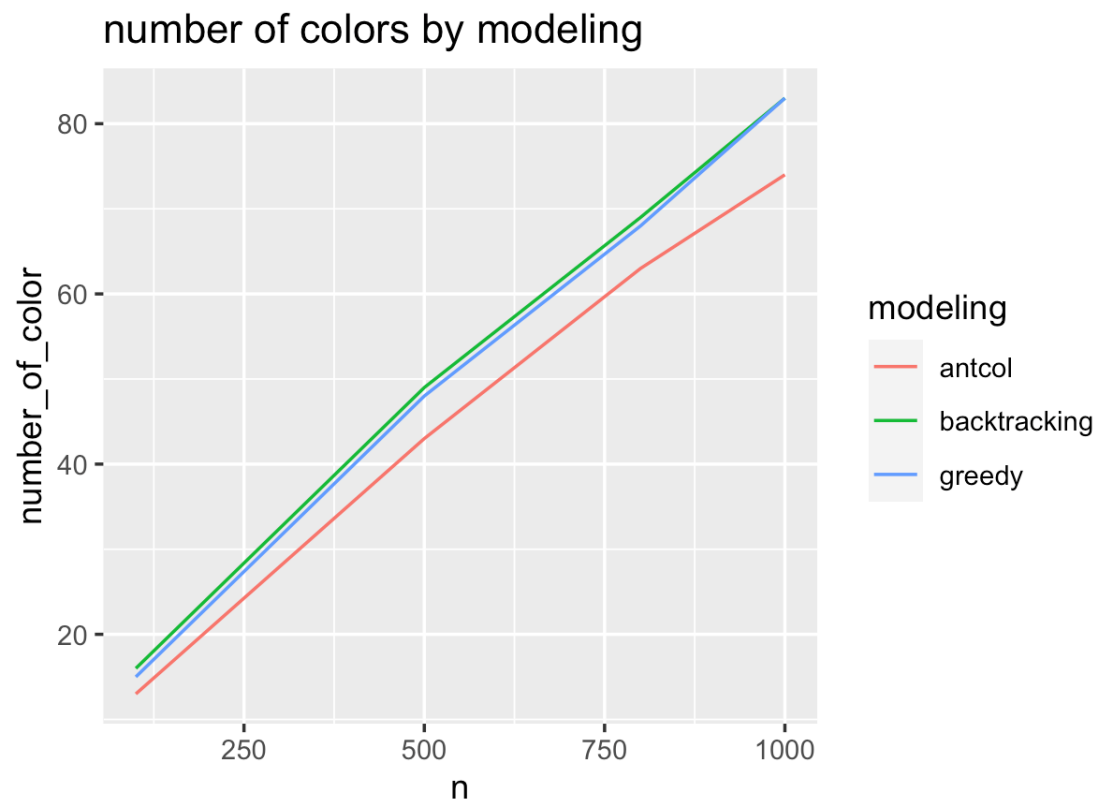
 $n = 1000, a = 4124$ 

	Greedy	ACO	Backtracking
Number of Colors	83	74	83
Computation time	6.2304 sec	12200.6978 sec	3.0254 sec

## 03. 시뮬레이션

### Simulation 3. 그래프 크기의 증가 ( $n, a$ )

그래프 크기 증가에 따른 비교 분석

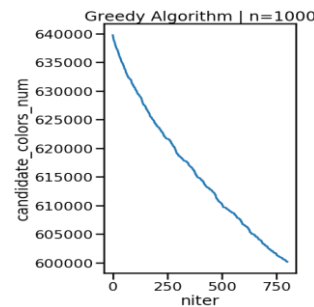
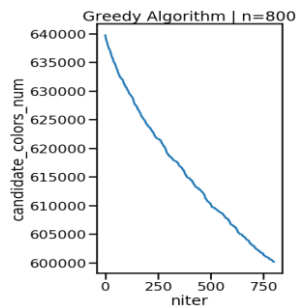
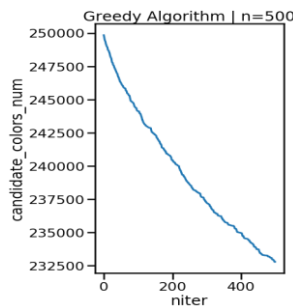
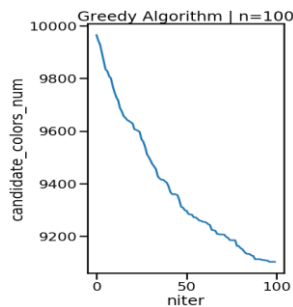


## 03. 시뮬레이션

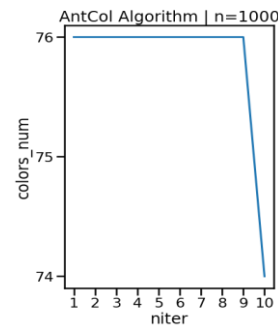
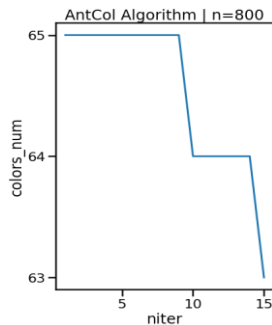
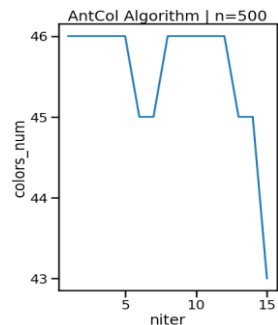
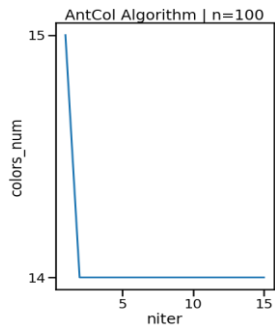
### Simulation 3. 그래프 크기의 증가 ( $n, a$ )

알고리즘 수렴 여부에 대한 비교

#### Greedy 알고리즘



#### AntCol 알고리즘



- 그래프 크기가 클 때, Greedy 와 AntCol 모두 수렴했다고 보기 어렵다. 개선의 여지가 있다.
- AntCol 이 Greedy 보다 안정적으로 수렴하는 경향이 있다.

## 03. 시뮬레이션

### Conclusion

최적의 값을 찾고 싶을 경우 → ACO 사용

적당한 값을 찾고 싶을 경우 → Greedy 사용

- ACO 알고리즘은 가장 최소의 color 수를 찾는다.  
하지만 ACO은 다른 알고리즘에 비해 상당히 까다롭다. 메모리 양과 계산 시간이 ACO에서는 개미와 페로몬에 할당된다.
- Greedy 알고리즘은 연산시간이 가장 짧다. 다른 알고리즘이 시간 단위로 걸리는 문제도 10초 이내로 color 수를 찾는다.  
하지만 Greedy로 찾은 color의 수는 최적의 color 수는 아니었다.

# 04

## 알고리즘 활용

Timetable Scheduling with Graph Coloring

## 04. 알고리즘 활용

### Timetable problem 이란?

- ▶ 제한조건 하에서 가능한 time slot에 활동 (수업, 업무일정 등)을 배분하는 것  
ex) 간호사 shift 시간표, 대학교 시험시간표
- ▶ <대학교 시험시간표>
  - vertex : 수업
  - edge : 겹치면 안 되는 수업들을 이은 선
- ▶ 제한조건의 종류
  - hard constraint : 무조건 만족해야 하는 조건
  - soft constraint : 만족하면 좋은 조건



## 04. 알고리즘 활용

### 이화여대 통계학과 시험/발표 시간표 배정

**why** 시험 시간이 보통 1시간 30분이 넘기 때문에 수업 시간 내에 치룰 수 없고 따로 모든 학생들이 가능한 시간을 배정하는 것은 힘들다.

**Goal** 통계학과 의 총 25개 수업을 최소 색깔(time slot)으로 구분하기

● 시험시간은 3시간으로 한다.

● 총 3개의 period    8:00 ~ 11:00    12:00 ~ 15:00    15:00 ~ 18:00

● 제한 조건 :

#### Hard

1. 교수님은 질의응답을 위해 각 시험에 방문하신다. (교수님이 같은 시간에 여러 시험장에 계실 수 없다.)
2. 데이터 과학자 트랙을 수강하는 학부생도 있다.  
(기초확률론, 회귀분석, 수리통계학2, 학부 데이터마이닝, 통계프로그래밍 필수. 즉, 이 과목들을 한번에 듣는 학생도 있으니 겹치면 안 된다.)

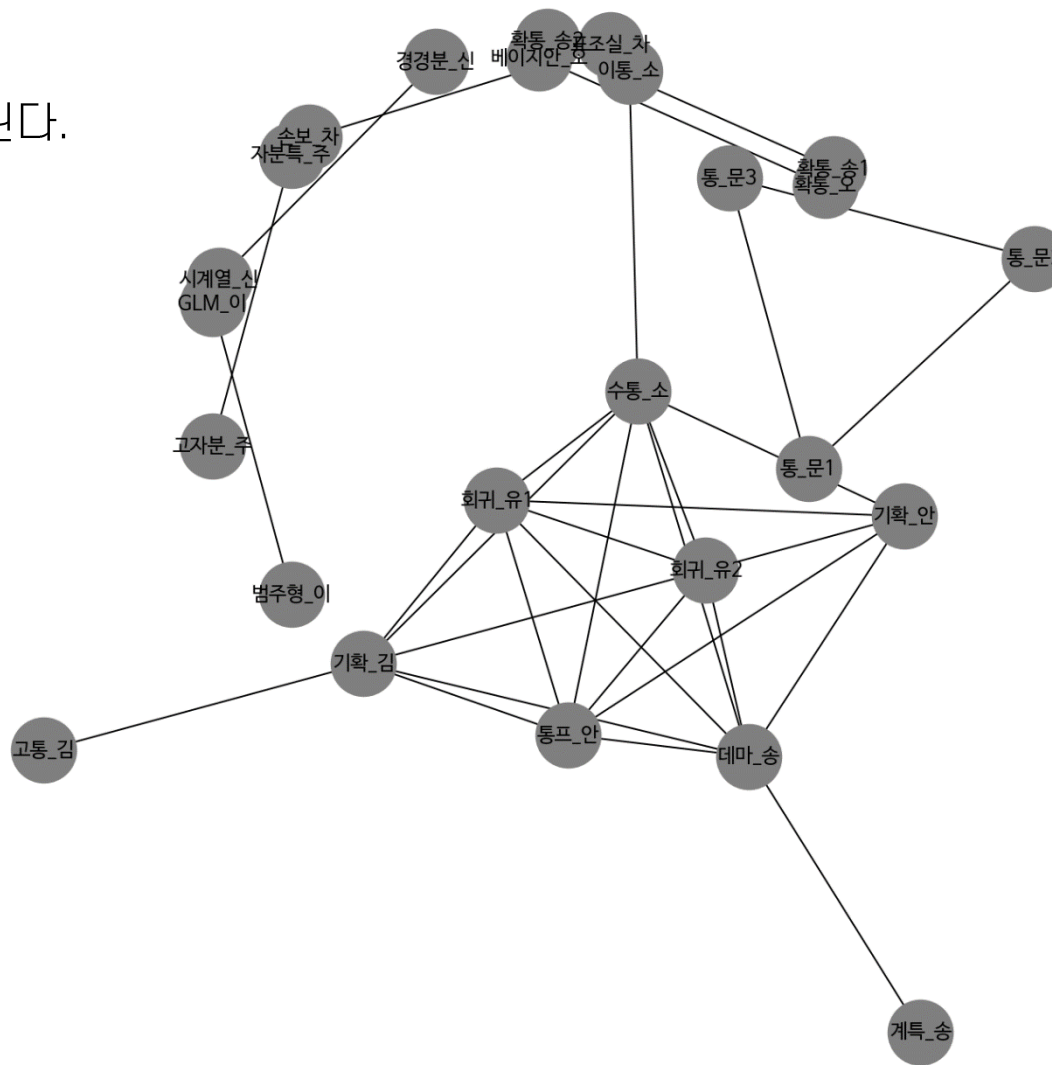
#### Soft

1. 점심시간에는 시험을 치지 않는다 (11:00 ~ 12:00)
2. 각 학년의 수업은 시험시간이 겹치지 않는 것이 좋다.
3. 원래 수업 요일에 시험을 치르면 좋다.

## 04. 알고리즘 활용

### Network Graph

- 동시간에 시험을 치를 수 없는 수업은 선이 연결된다.
- 총 25개 vertices  
32개 edges



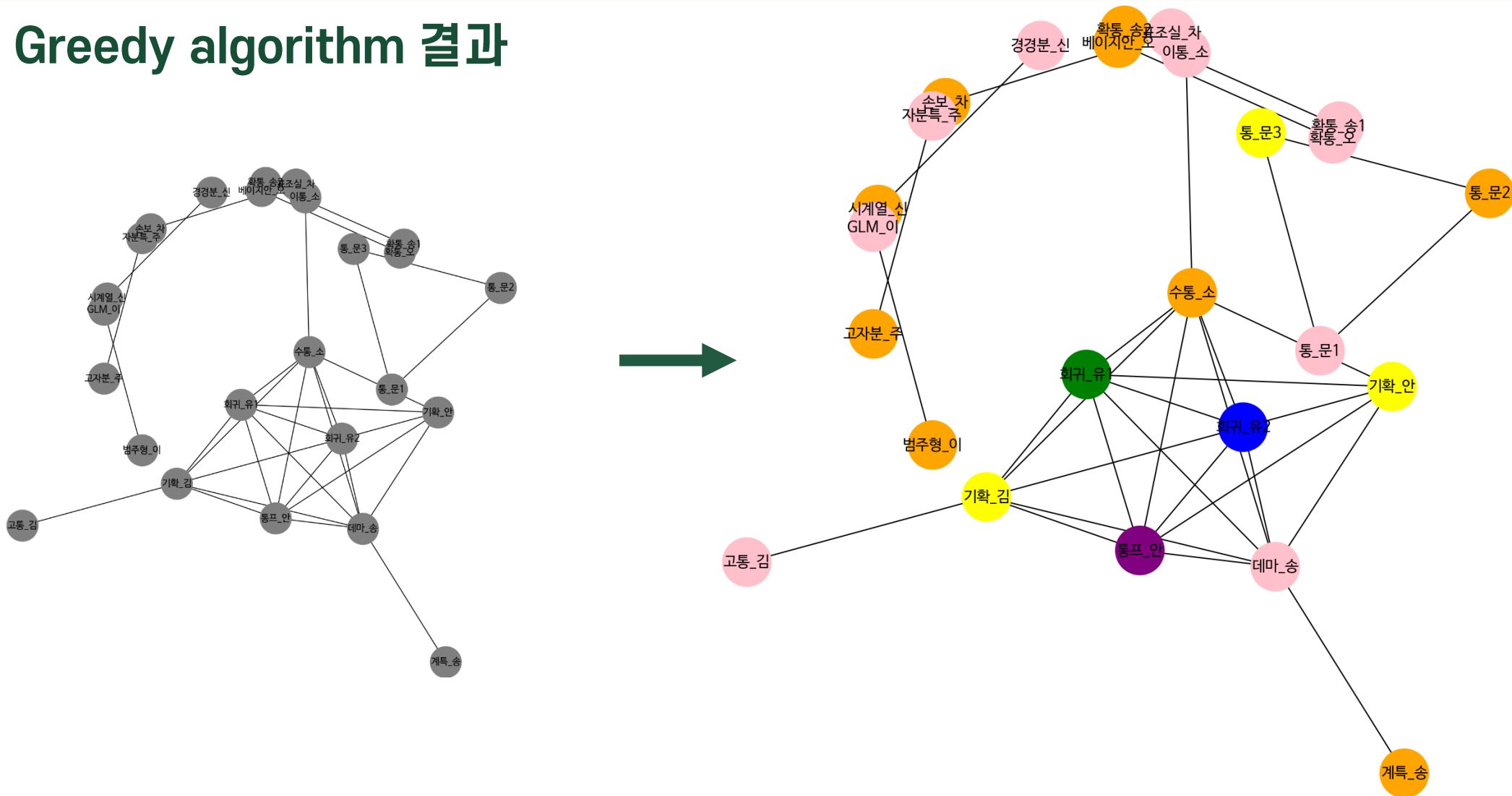
## 04. 알고리즘 활용

## 시간표 배정 결과

Algorithm	System time	색의 개수	결과 (25개 수업명을 알파벳으로 표시)
greedy	0.028	6	색 0 : [a, d, e, h, j, l, n, p, v, y] 색 1 : [b, f, g, i, k, m, o, w, x] 색 2 : [c, q, r] 색 3 : [s] 색 4 : [t] 색 5 : [u]
AntCol	0.3251	6	색 0 : [a, d, f, h, j, l, n, p, v, y] 색 1 : [b, e, g, i, k, m, o, q, r, w] 색 2 : [c, t] 색 3 : [x] 색 4 : [u] 색 5 : [s]
backtracking	0.0168	7	색 0 : [a, d, e, h, j, l, n, p, r, w, y] 색 1 : [b, f, g, i, k, m, o, q] 색 2 : [c, s] 색 3 : [t] 색 4 : [u] 색 5 : [v] 색 6 : [x]

## 04. 알고리즘 활용

### 1. Greedy algorithm 결과



## 04. 알고리즘 활용

### 1. Greedy algorithm

### Step 1 : graph coloring 결과

다른 색이 칠해진 수업은 동시간에 시험을 치를 수 없다.

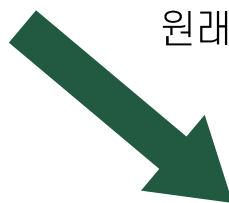
Color 0	Color 1	Color 2	Color 3	Color 4	Color 5
(a) 통2 1분반 (문온교수님)	(b) 통2 2분반 (문온교수님)	(c) 통2 3분반 (문온교수님)	(s) 회귀분석1분반(유재근교수님)	(t) 회귀분석2분반(유재근교수님)	(u) 통계프로그래밍(안재윤교수님)
(d) 확통 1분반 (오만숙교수님)	(f) 확통 3분반 (송수민교수님)	(q) 기초확률론(김경원교수님)			
(e) 확통 2분반 (송수민교수님)	(g) 베이지안 (오만숙교수님)	(r) 기초확률론 (안재윤교수님)			
(h) 표조실 (차지환교수님)	(i) 손해보험론 (차지환교수님)				
(j) 경자분 (신동완교수님)	(k) 시계열 (신동완교수님)				
(l) 자분특 (주원영교수님)	(m) 고급자료분석(주원영교수님)				
(n) GLM (이동환교수님)	(o) 범주형 (이동환교수님)				
(p) 고급통계 (김경원교수님)	(w) 통계계산특론(송종우교수님)				
(v) 데이터마이닝(송종우교수님)	(x) 수리통계2 (소병수교수님)				
(y) 이론통계2 (소병수교수님)					

## 04. 알고리즘 활용

### 1. Greedy algorithm

#### Step 2 : soft 제한 조건 고려하기 (1학년)

1학년	수업명
8:00 ~ 11:00	확통 1분반(오만숙교수님) 확통 2분반 (송수민교수님) 통2 1분반 (문온교수님)
11:00 ~ 12:00	Lunch
12:00 ~ 15:00	통2 2분반 (문온교수님) 확통 3분반 (송수민교수님)
15:00 ~ 18:00	통2 2분반 (문온교수님)



원래 수업 요일을 고려하여 최대한 맞춰 나누기

1학년	월	화	수	목	금
8:00 ~ 11:00		확통 1분반 (오만숙교수님)	통계학2 1분반 (문온교수님)	확통 2분반 (송수민교수님)	
11:00 ~ 12:00	Lunch				
12:00 ~ 15:00	통계학2 2분반 (문온교수님)	확통 3분반 (송수민교수님)			
15:00 ~ 18:00	통계학2 3분반 (문온교수님)				

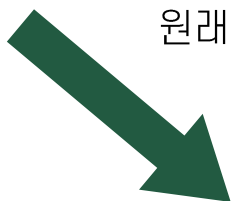
## 04. 알고리즘 활용

### 1. Greedy algorithm

#### Step 2 : soft 제한 조건 고려하기 (2학년)

2학년	수업명
8:00 ~ 11:00	회귀 1분반(유재근교수님)
11:00 ~ 12:00	Lunch
12:00 ~ 15:00	회귀 2분반(유재근교수님)
15:00 ~ 18:00	기초확률론(안재윤교수님) 기초확률론(김경원교수님)

원래 수업 요일을 고려하여 최대한 맞춰 나누기



2학년	월	화	수	목	금
8:00 ~ 11:00					회귀분석 1분반 (유재근교수님)
11:00 ~ 12:00	Lunch				
12:00 ~ 15:00					회귀분석 2분반 (유재근교수님)
15:00 ~ 18:00	기초확률론 (안재윤교수님)	기초확률론 (김경원교수님)			

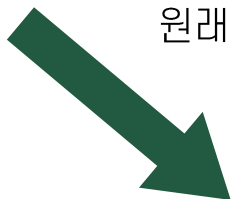
## 04. 알고리즘 활용

### 1. Greedy algorithm

#### Step 2 : soft 제한 조건 고려하기 (3학년)

3학년	수업명
8:00 ~ 11:00	데이터마이닝(송종우교수님) 표본조사실습(차지환교수님)
11:00 ~ 12:00	Lunch
12:00 ~ 15:00	수리통계학2(소병수교수님)
15:00 ~ 18:00	통계프로그래밍(안재윤교수님)

원래 수업 요일을 고려하여 최대한 맞춰 나누기



3학년	월	화	수	목	금
8:00 ~ 11:00	표본조사실습 (차지환교수님)			데이터마이닝 (송종우교수님)	
11:00 ~ 12:00	Lunch				
12:00 ~ 15:00		수리통계학2 (소병수교수님)			
15:00 ~ 18:00				통계프로그래밍 (안재윤교수님)	



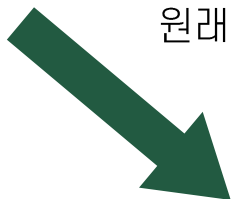
## 04. 알고리즘 활용

### 1. Greedy algorithm

#### Step 2 : soft 제한 조건 고려하기 (4학년)

4학년	수업명
8:00 ~ 11:00	경자분 (신동완교수님)
11:00 ~ 12:00	Lunch
12:00 ~ 15:00	범주형 (이동환교수님) 고급자료분석 (주원영교수님)
15:00 ~ 18:00	

원래 수업 요일을 고려하여 최대한 맞춰 나누기



4학년	월	화	수	목	금
8:00 ~ 11:00		경영경제자료분석 (신동완교수님)			
11:00 ~ 12:00	Lunch				
12:00 ~ 15:00	고급자료분석 (주원영교수님)		범주형자료분석 (이동환교수님)		
15:00 ~ 18:00					

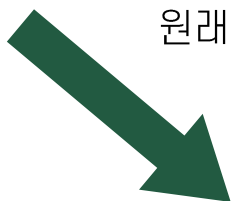
## 04. 알고리즘 활용

### 1. Greedy algorithm

#### Step 2 : soft 제한 조건 고려하기 (대학원생)

대학원생	수업명
8:00 ~ 11:00	고급통계 (김경원교수님) 이론통계2 (소병수교수님) GLM 이동환교수님 자료분석특론 (주원영교수님)
11:00 ~ 12:00	Lunch
12:00 ~ 15:00	통계특 (송종우교수님) 시계열 (신동완교수님) 베이지안 (오만숙교수님) 손해보험론 (차지환교수님)
15:00 ~ 18:00	

원래 수업 요일을 고려하여 최대한 맞춰 나누기



대학원생	월	화	수	목	금
8:00 ~ 11:00	고급통계학 (김경원교수님)	GLM (이동환교수님)	이론통계2 (소병수교수님)	자료분석특론 (주원영교수님)	
11:00 ~ 12:00	Lunch				
12:00 ~ 15:00	통계계산특론 (송종우교수님)	시계열 (신동완교수님)	손해보험론 (차지환교수님)	베이지안통계 (오만숙교수님)	
15:00 ~ 18:00					

## 04. 알고리즘 활용

### 1. Greedy algorithm

## 김이화 2022-2 기말 시험/프로젝트 시간표

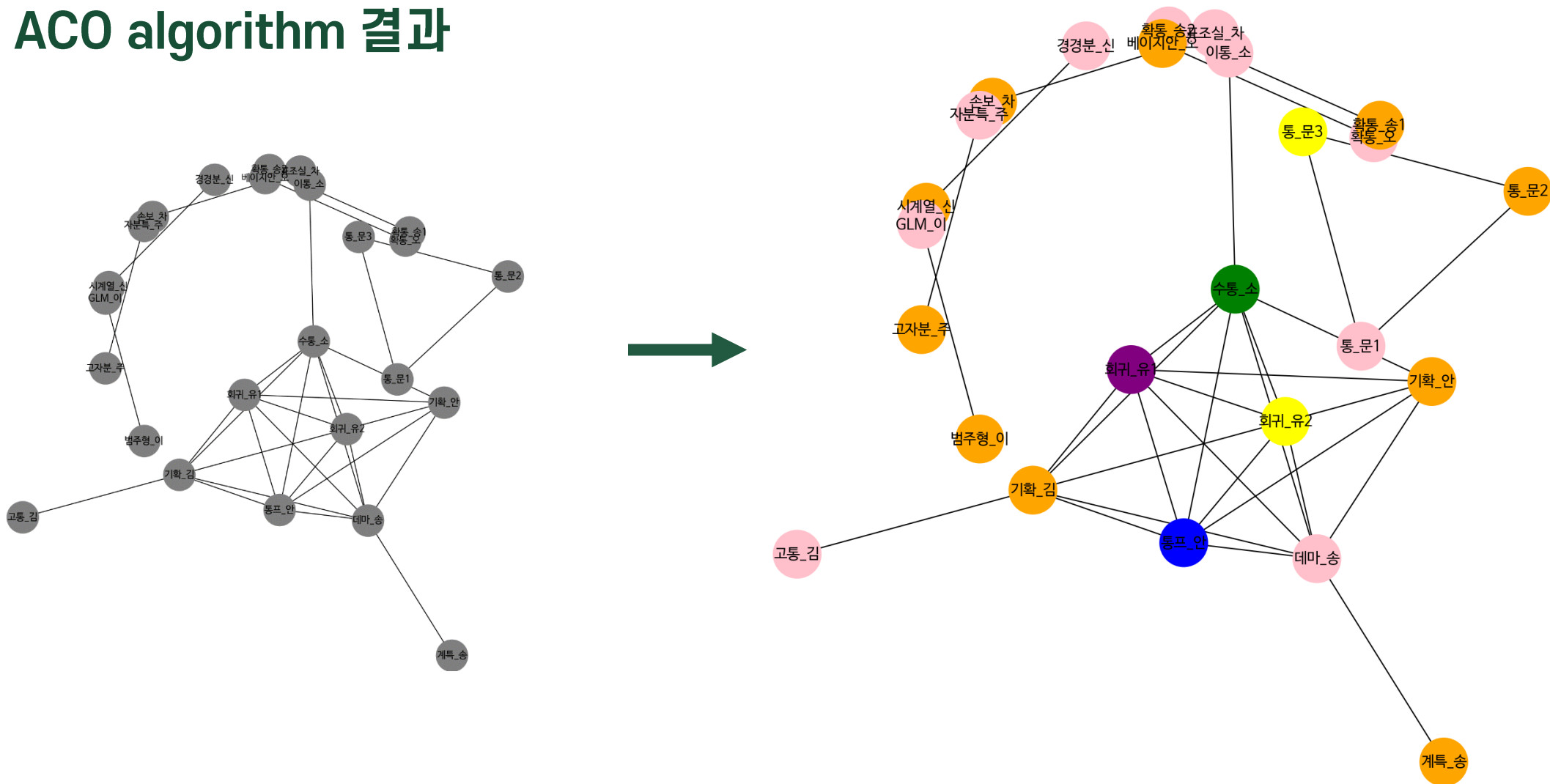
- 데이터 과학자 트랙을 신청한 학부생
- 이번학기에 기초확률론, 회귀분석, 수리통계2, 데이터마이닝, 통계프로그래밍을 수강한다.

### 시험 시간표 (12/19 ~ 12/23)

김이화	월	화	수	목	금
8:00 ~ 11:00				데이터마이닝 (송종우교수님)	회귀분석 1분반 (유재근교수님)
11:00 ~ 12:00	Lunch				
12:00 ~ 15:00		수리통계학2 (소병수교수님)			
15:00 ~ 18:00	기초확률론 (안재윤교수님)			통계프로그래밍 (안재윤교수님)	

## 04. 알고리즘 활용

### 2. ACO algorithm 결과



## 04. 알고리즘 활용

### 2. ACO algorithm

#### Step 1 : graph coloring 결과

다른 색이 칠해진 수업은 동시간에 시험을 치를 수 없다.

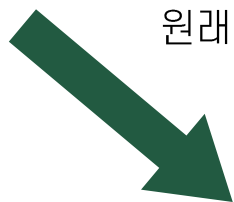
Color 0	Color 1	Color 2	Color 3	Color 4	Color 5
(a) 통계학 1분반 (문온교수님)	(b) 통계학 2분반 (문온교수님)	(c) 통계학 3분반 (문온교수님)	(x) 수리통계학2(소병수교수님)	(u) 통계프로그래밍(안재윤교수님)	(s) 회귀분석1분반(유재근교수님)
(d) 확통 1분반 (오만숙교수님)	(e) 확통 2분반 (송수민교수님)	(t) 회귀분석2분반(유재근교수님)			
(f) 확통 3분반 (오만숙교수님)	(g) 베이지안 (오만숙교수님)				
(h) 표본조사및실습 (차지환교수님)	(i) 손해보험론 (차지환교수님)				
(j) 경자분 (신동완교수님)	(k) 시계열 (신동완교수님)				
(l) 자료분석특론(주원영교수님)	(m) 고급자료분석(주원영교수님)				
(n) GLM (이동환교수님)	(o) 범주형 (이동환교수님)				
(p) 고급통계 (김경원교수님)	(q) 기초확률론 (김경원교수님)				
(v) 데이터마이닝(송종우교수님)	(r) 기초확률론 (안재윤교수님)				
(y) 이론통계2 (소병수교수님)	(w) 통계계산특론(송종우교수님)				

## 04. 알고리즘 활용

### 2. ACO algorithm

#### Step 2 : soft 제한 조건 고려하기 (1학년)

1학년	수업명
8:00 ~ 11:00	확통 1분반(오만숙교수님) 확통 2분반 (송수민교수님) 통2 1분반 (문온교수님)
11:00 ~ 12:00	Lunch
12:00 ~ 15:00	통2 2분반 (문온교수님) 확통 3분반 (송수민교수님)
15:00 ~ 18:00	통2 2분반 (문온교수님)



원래 수업 요일을 고려하여 최대한 맞춰 나누기

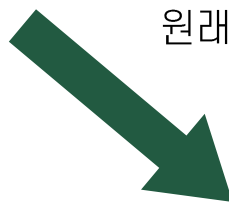
1학년	월	화	수	목	금
8:00 ~ 11:00	확통 1분반 (오만숙교수님)	확통 3분반 (송수민교수님)	통계학 1분반 (문온교수님)		
11:00 ~ 12:00	Lunch				
12:00 ~ 15:00	통계학2 2분반 (문온교수님)	확통 3분반 (송수민교수님)			
15:00 ~ 18:00	통계학2 3분반 (문온교수님)				

## 04. 알고리즘 활용

### 2. ACO algorithm

#### Step 2 : soft 제한 조건 고려하기 (2학년)

2학년	수업명
8:00 ~ 11:00	회귀분석1분반(유재근교수님)
11:00 ~ 12:00	Lunch
12:00 ~ 15:00	기초확률론 (안재윤교수님) 기초확률론 (김경원교수님)
15:00 ~ 18:00	회귀분석2분반(유재근교수님)



원래 수업 요일을 고려하여 최대한 맞춰 나누기

2학년	월	화	수	목	금
8:00 ~ 11:00				회귀분석1분반 (유재근교수님)	
11:00 ~ 12:00	Lunch				
12:00 ~ 15:00	기초확률론 (안재윤교수님)	기초확률론 (김경원교수님)			
15:00 ~ 18:00			회귀분석2분반 (유재근교수님)		

## 04. 알고리즘 활용

### 2. ACO algorithm

#### Step 2 : soft 제한 조건 고려하기 (3학년)

3학년	수업명
8:00 ~ 11:00	데이터마이닝 (송종우교수님) 표본조사및실습(차지환교수님)
11:00 ~ 12:00	Lunch
12:00 ~ 15:00	수리통계학2 (소병수교수님)
15:00 ~ 18:00	통계프로그래밍 (안재윤교수님)

원래 수업 요일을 고려하여 최대한 맞춰 나누기

3학년	월	화	수	목	금
8:00 ~ 11:00	표본조사및실습 (차지환교수님)	데이터마이닝 (송종우교수님)			
11:00 ~ 12:00	Lunch				
12:00 ~ 15:00				수리통계학2 (소병수교수님)	
15:00 ~ 18:00				통계프로그래밍 (안재윤교수님)	

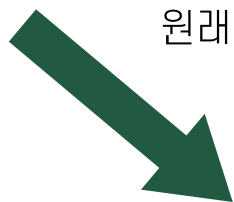


## 04. 알고리즘 활용

### 2. ACO algorithm

#### Step 2 : soft 제한 조건 고려하기 (4학년)

4학년	수업명
8:00 ~ 11:00	경자분 (신동완교수님)
11:00 ~ 12:00	Lunch
12:00 ~ 15:00	범주형 (이동환교수님) 고급자료분석 (주원영교수님)
15:00 ~ 18:00	



원래 수업 요일을 고려하여 최대한 맞춰 나누기

4학년	월	화	수	목	금
8:00 ~ 11:00		경영경제자료분석 (신동완교수님)			
11:00 ~ 12:00	Lunch				
12:00 ~ 15:00	고급자료분석 (주원영교수님)		범주형자료분석 (이동환교수님)		
15:00 ~ 18:00					


## 04. 알고리즘 활용

### 2. ACO algorithm

#### Step 2 : soft 제한 조건 고려하기 (대학원생)

대학원생	수업명
8:00 ~ 11:00	고급통계 (김경원교수님) 이론통계2 (소병수교수님) GLM 이동환교수님) 자료분석특론 (주원영교수님)
11:00 ~ 12:00	Lunch
12:00 ~ 15:00	통계특 (송종우교수님) 시계열 (신동완교수님) 베이지안 (오만숙교수님) 손해보험론 (차지환교수님)
15:00 ~ 18:00	

원래 수업 요일을 고려하여 최대한 맞춰 나누기



대학원생	월	화	수	목	금
8:00 ~ 11:00	고급통계학 (김경원교수님)	GLM (이동환교수님)	이론통계2 (소병수교수님)	자료분석특론 (주원영교수님)	
11:00 ~ 12:00	Lunch				
12:00 ~ 15:00	통계계산특론 (송종우교수님)	시계열 (신동완교수님)	손해보험론 (차지환교수님)	베이지안통계 (오만숙교수님)	
15:00 ~ 18:00					

## 04. 알고리즘 활용

### 2. ACO algorithm

## 김이화 2022-2 기말 시험/프로젝트 시간표

- 데이터 과학자 트랙을 신청한 학부생
- 이번학기에 기초확률론, 회귀분석, 수리통계2, 데이터마이닝, 통계프로그래밍을 수강한다.

### 시험 시간표 (12/19 ~ 12/23)

김이화	월	화	수	목	금
8:00 ~ 11:00		데이터마이닝 (송종우교수님)			
11:00 ~ 12:00	Lunch				
12:00 ~ 15:00	기초확률론 (안재윤교수님)			수리통계학2 (소병수교수님)	
15:00 ~ 18:00			회귀분석 2분반 (유재근교수님)	통계프로그래밍 (안재윤교수님)	

## 참고문헌

---

- Comparative Analysis of the main Graph Coloring Algorithms (2021)
- Analysis of a parallel MCMC algorithm for graph coloring with nearly uniform balancing
- A study on Course Timetable Scheduling Using Graph Coloring Approach
- A Graph Coloring Algorithm for Large Scheduling Problems(1976)

# 감사합니다

---

212STG34 강채리

212STG35 노예림

212STG36 백나림

212STG39 유찬미