알고리즘과 코딩테스트 준비

2022/1/13~1/19

윤형기 (hky@openwith.net)

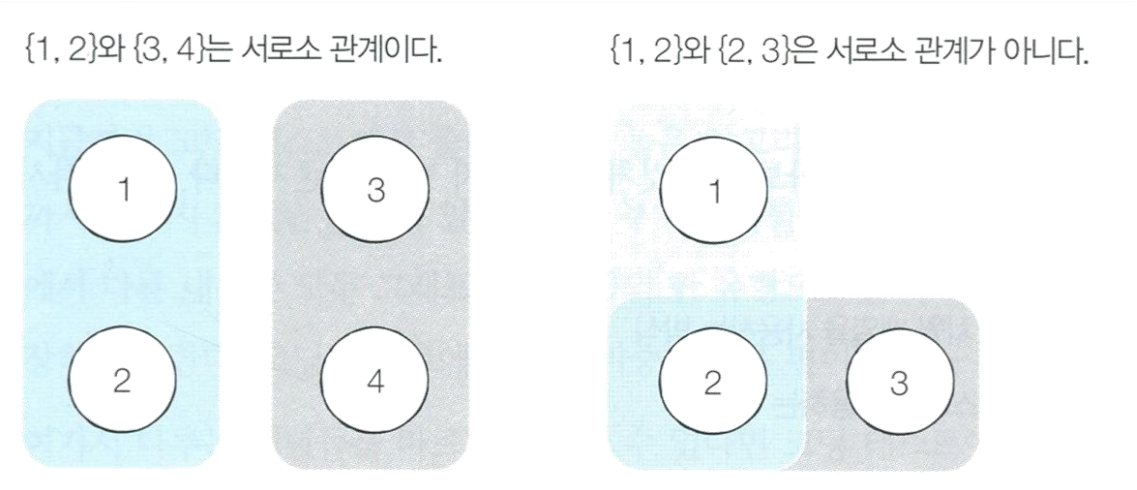
그래프 (3) Chapter 10 그래프 이론

- 그래프 vs. Tree

	그래프	Tree
방향성	방향 그래프 or 무방향 그래프	방향 그래프
순환성	순환 및 비순환	비순환
노드간 관계성	부모와 자식관계 없음	부모와 자식관계
모델의 종류	네트워크 모델	계층모델

서로소 집합

- 서로소 집합 자료구조 (Disjoint Sets)
  - 서로소 부분집합들로 나누어진 원소들의 데이터를 처리하기 위한 데이터 구조 = Union Find 자료구조
- 두 종류의 연산을 지원
  - 합집합 (Union): 두 원소가 포함된 집합을 한 집합으로 합치는 것
  - 찾기 (Find): 특정 원소가 속한 집합이 어떤 집합인지 알려주는 연산

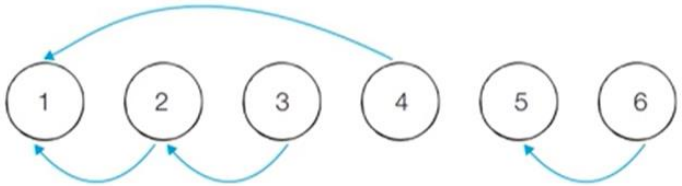


- 서로소 집합계산 알고리즘
  - (1) Union 연산을 확인하여 서로 연결된 두 node A,B를 확인
    - a. A와 B의 root node A', B'을 각각 찾는다
    - b. A'을 B'의 부모노드로 설정한다.
  - 모든 합집합 (Union) 연산을 처리할 때까지 1번의 과정을 반복

- Union연산을 그래프로 표현
  - Tree구조를 Edge로 표현하는 것

- 예:
  - {1,2,3,4,5,6}이 있을 때 union {1,4}, {2,3},{2,4},{5,6}이 주어진다면 union연산 후 전체원소들은 어떤 부분집합으로 나누어지는가?

노드 번호	1	2	3	4	5	6
부모	1	2	3	4	5	6

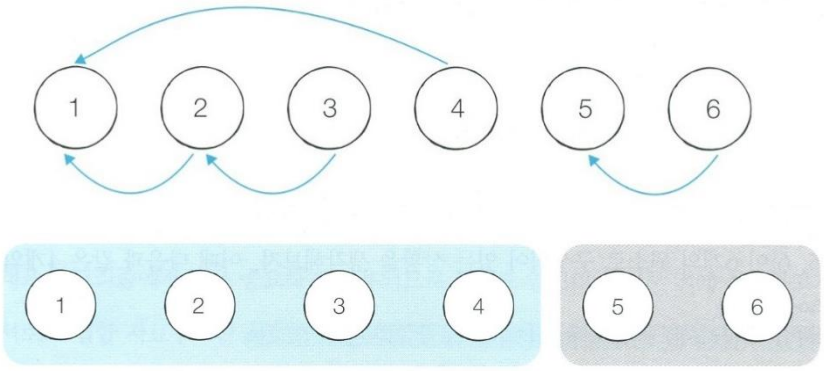


노드 번호	1	2	3	4	5	6
부모	1	1	2	1	5	5

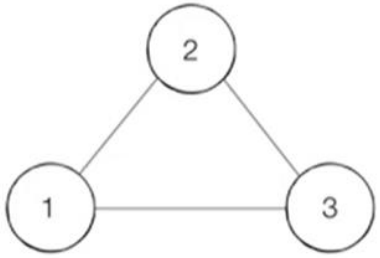
일반적으로 서로소 집합을 그림으로 표현 시
번호가 큰 node --> 번호가 작은 node의 tree구조로 표현

소스코드: 10-1 > (path compression) 10-2, 10-3

- 결과적으로



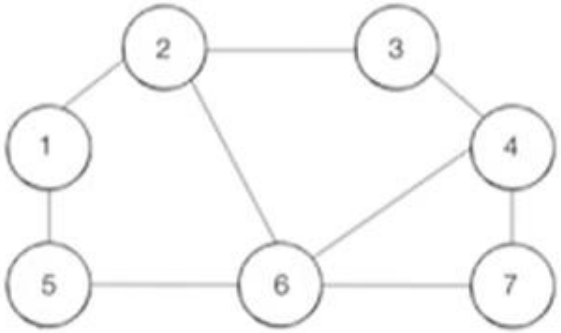
- 서로소 집합을 활용한 사이클 판별
  - 무방향 그래프 내에서의 cycle판별에 이용
    - (cf. 방향 그래프 내에서의 cycle판별에는 DFS 이용)
  - Edge를 하나씩 확인 (즉, 두 node의 root node를 확인)
    - Root node가 다르다면 두 node에 대한 union연산 수행
    - Root node가 같다면 cycle이 발생했음을 의미
  - (반복수행)
  - 예:



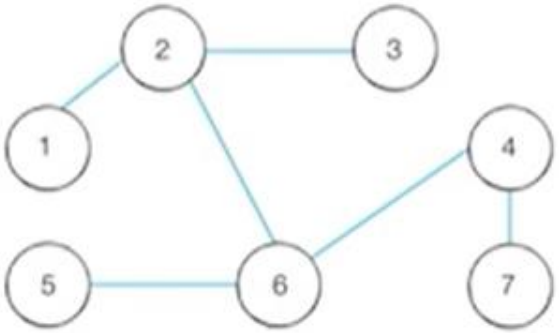
인덱스	1	2	3
부모	1	2	3

신장트리

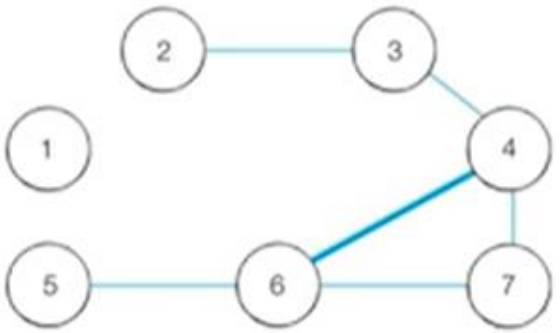
- 신장트리 (Spanning Tree)란?
  - 모든 노드를 포함하면서 사이클이 존재하지 않는 부분 그래프
    - 이는 Tree의 조건이기도 함



원본 그래프

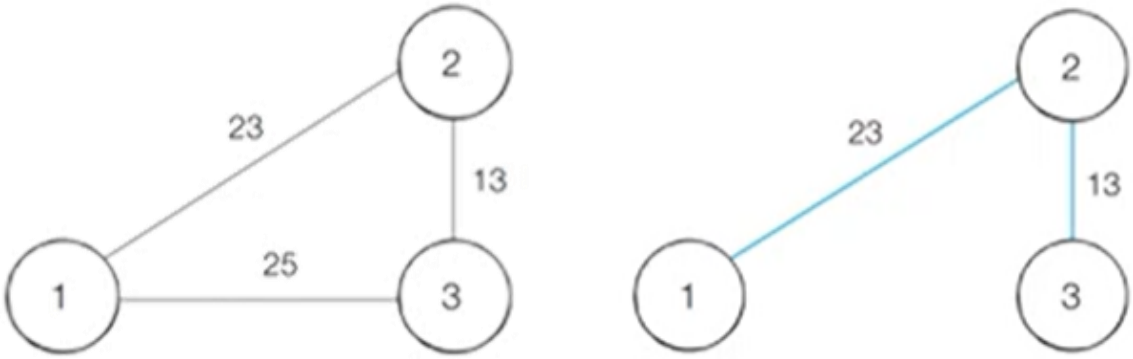


가능한 신장 트리 예시

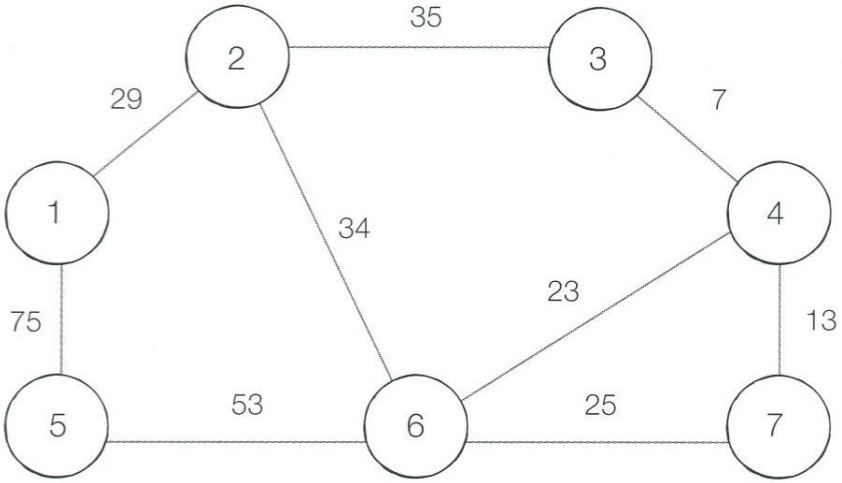


신장 트리가 아닌 부분 그래프 예시

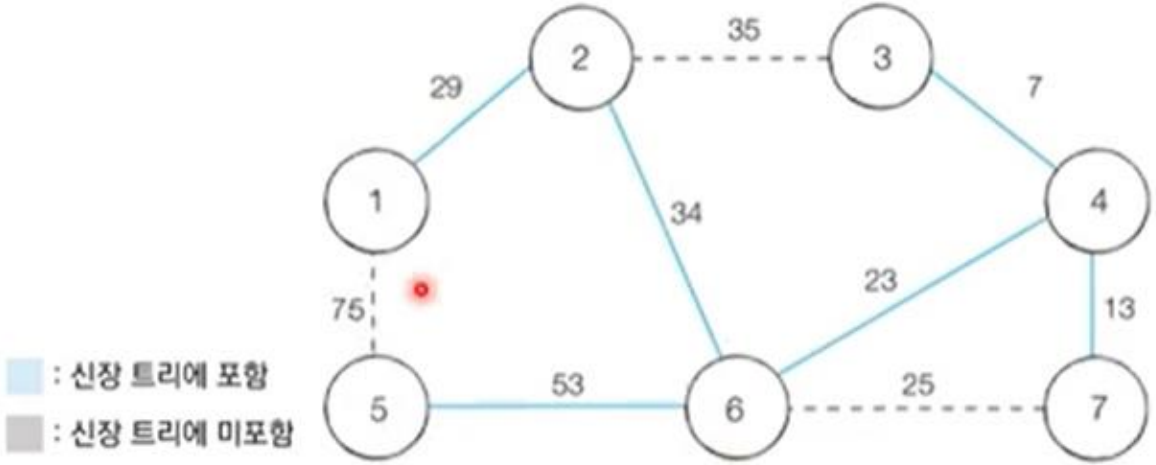
- Kruskal 알고리즘
  - 최소비용으로 Spanning Tree를 찾음
    - 예: 두 도시를 도로로 연결



  - 알고리즘
    - 1. 간선 데이터를 비용에 따라 오름차순으로 정렬
    - 2. 간선을 하나씩 확인하며 현재 간선이 cycle을 발생시키는지 확인
      - A. Cycle이 발생하지 않는 경우 최소신장트리에 포함시킴
      - B. Cycle이 발생하는 경우 최소신장트리에 포함시키지 않음
    - 3. 모든 간선에 대해 2번의 과정을 반복

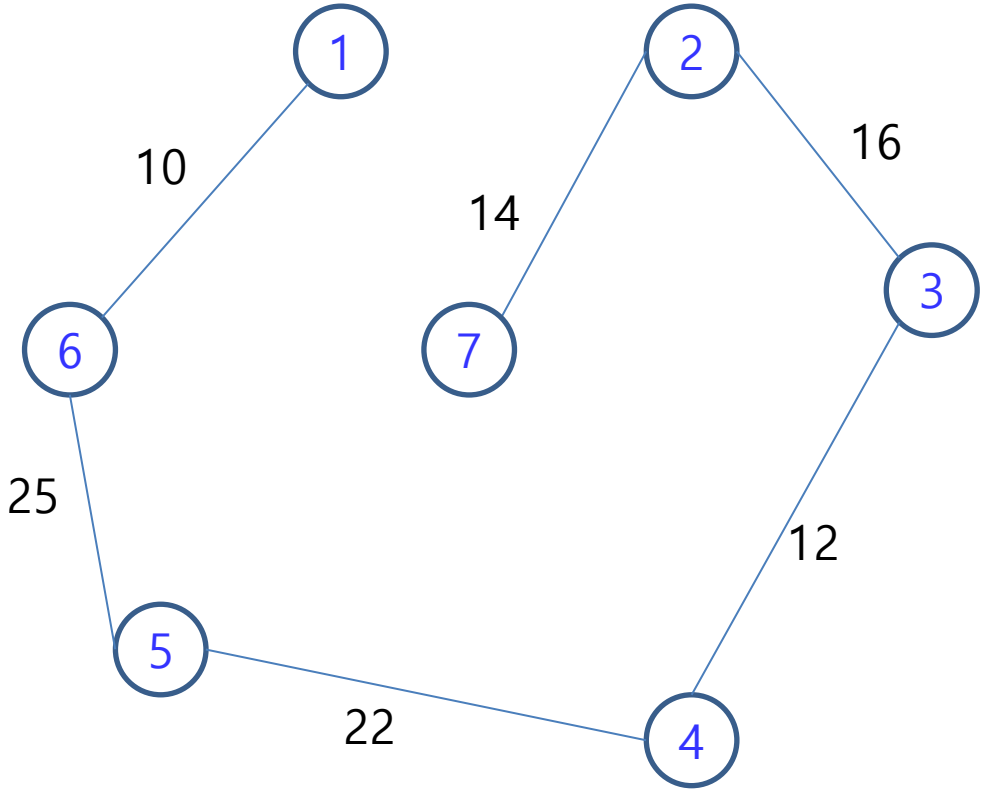


간선	(1, 2)	(1, 5)	(2, 3)	(2, 6)	(3, 4)	(4, 6)	(4, 7)	(5, 6)	(6, 7)
비용	29	75	35	34	7	23	13	53	25



간선	(1, 2)	(1, 5)	(2, 3)	(2, 6)	(3, 4)	(4, 6)	(4, 7)	(5, 6)	(6, 7)
비용	29	75	35	34	7	23	13	53	25
순서	step 5	step 9	step 7	step 6	step 1	step 3	step 2	step 8	step 4

- Prim's algorithm
  - Select the minimum cost edge first! 단, cycle회피!

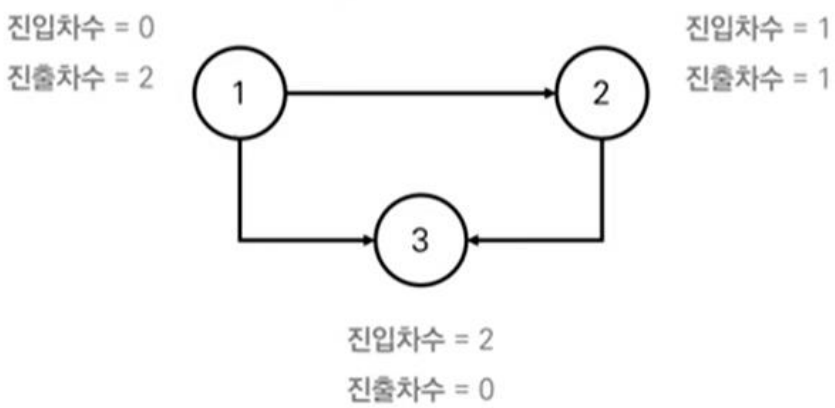


위상정렬

- 위상 정렬 (Topology Sort)
  - Cycle이 없는 방향 그래프의 모든 노드를 방향성에 거스르지 않도록 순서대로 나열하는 것
  - 예: 선수과목을 고려한 학습순서 설정

- 위 세 과목을 모두 듣기 위한 **적절한 학습 순서**는?
  - 자료구조 → 알고리즘 → 고급 알고리즘 (O)
  - 자료구조 → 고급 알고리즘 → 알고리즘 (X)

- 진입차수와 진출차수
  - 진입차수 (Indegree)
  - 진출차수 (Outdegree)



- 특징
  - 위상정렬은 DAG에 대해서만 수행 가능
  - 위상정렬에서는 여러 가지 답이 존재할 수 있음.
    - 한 단계에서 queue에 새롭게 들어가는 원소가 2개 이상인 경우가 있다면 여러 가지 답이 존재
  - Stack을 활용한 DFS를 이용해 위상정렬을 수행할 수도 있음.

2 [실전 문제] 팀 결성

난이도 ●●○ | 풀이 시간 20분 | 시간 제한 2초 | 메모리 제한 128MB | 기출 핵심 유형

학교에서 학생들에게 0번부터 N번까지의 번호를 부여했다. 처음에는 모든 학생이 서로 다른 팀으로 구분되어, 총 N + 1개의 팀이 존재한다. 이때 선생님은 '팀 합치기' 연산과 '같은 팀 여부 확인' 연산을 사용할 수 있다.

1. '팀 합치기' 연산은 두 팀을 합치는 연산이다.
2. '같은 팀 여부 확인' 연산은 특정한 두 학생이 같은 팀에 속하는지를 확인하는 연산이다.

선생님이 M개의 연산을 수행할 수 있을 때, '같은 팀 여부 확인' 연산에 대한 연산 결과를 출력하는 프로그램을 작성하시오.

입력 조건

- 첫째 줄에 N, M이 주어진다. M은 입력으로 주어지는 연산의 개수이다. ($1 \le N, M \le 100,000$)
- 다음 M개의 줄에는 각각의 연산이 주어진다.
- '팀 합치기' 연산은 0 a b 형태로 주어진다. 이는 a번 학생이 속한 팀과 b번 학생이 속한 팀을 합친다는 의미이다.
- '같은 팀 여부 확인' 연산은 1 a b 형태로 주어진다. 이는 a번 학생과 b번 학생이 같은 팀에 속해 있는지를 확인하는 연산이다.
- a와 b는 N 이하의 양의 정수이다.

출력 조건

- '같은 팀 여부 확인' 연산에 대하여 한 줄에 하나씩 YES 혹은 NO로 결과를 출력한다.

입력 예시

```
7 8
0 1 3
1 1 7
0 7 6
1 7 1
0 3 7
0 4 2
0 1 1
1 1 1
```

출력 예시

```
NO
NO
YES
```

3 [실전 문제] 도시 분할 계획

난이도 ●●○ | 풀이 시간 40분 | 시간 제한 2초 | 메모리 제한 256MB | 기출 기초 문제집
출처 https://www.acmicpc.net/problem/1647

동물원에서 막 탈출한 원숭이 한 마리가 세상 구경을 하고 있다. 어느 날 원숭이는 '평화로운 마을'에 잠시 머물렀는데 마침 마을 사람들은 도로 공사 문제로 머리를 맞대고 회의 중이었다.

마을은 N개의 집과 그 집들을 연결하는 M개의 길로 이루어져 있다. 길은 어느 방향으로든지 다닐 수 있는 편리한 길이다. 그리고 길마다 길을 유지하는데 드는 유지비가 있다.

마을의 이장은 마을을 2개의 분리된 마을로 분할할 계획을 세우고 있다. 마을이 너무 커서 혼자서는 관리할 수 없기 때문이다. 마을을 분할할 때는 각 분리된 마을 안에 집들이 서로 연결되도록 분할해야 한다. 각 분리된 마을 안에 있는 임의의 두 집 사이에 경로가 항상 존재해야 한다는 뜻이다. 마을에는 집이 하나 이상 있어야 한다.

그렇게 마을의 이장은 계획을 세우다가 마을 안에 길이 너무 많다는 생각을 하게 되었다. 일단 분리된 두 마을 사이에 있는 길들은 필요가 없으므로 없앨 수 있다. 그리고 각 분리된 마을 안에서도 임의의 두 집 사이에 경로가 항상 존재하게 하면서 길을 더 없앨 수 있다. 마을의 이장은 위 조건을 만족하도록 길들을 모두 없애고 나머지 길의 유지비의 합을 최소로 하고 싶다. 이것을 구하는 프로그램을 작성하시오.

입력 조건

- 첫째 줄에 집의 개수 N, 길의 개수 M이 주어진다. N은 2 이상 100,000 이하인 정수이고, M은 1 이상 1,000,000 이하인 정수이다.
- 그다음 줄부터 M줄에 걸쳐 길의 정보가 A, B, C 3개의 정수로 공백으로 구분되어 주어지는데 A번 집과 B번 집을 연결하는 길의 유지비가 C($1 \le C \le 1,000$)라는 뜻이다.

출력 조건

- 첫째 줄에 길을 없애고 남은 유지비 합의 최솟값을 출력한다.

입력 예시

```
7 12
1 2 3
1 3 2
3 2 1
2 5 2
3 4 4
7 3 6
5 1 5
1 6 2
6 4 1
6 5 3
4 5 3
6 7 4
```

출력 예시

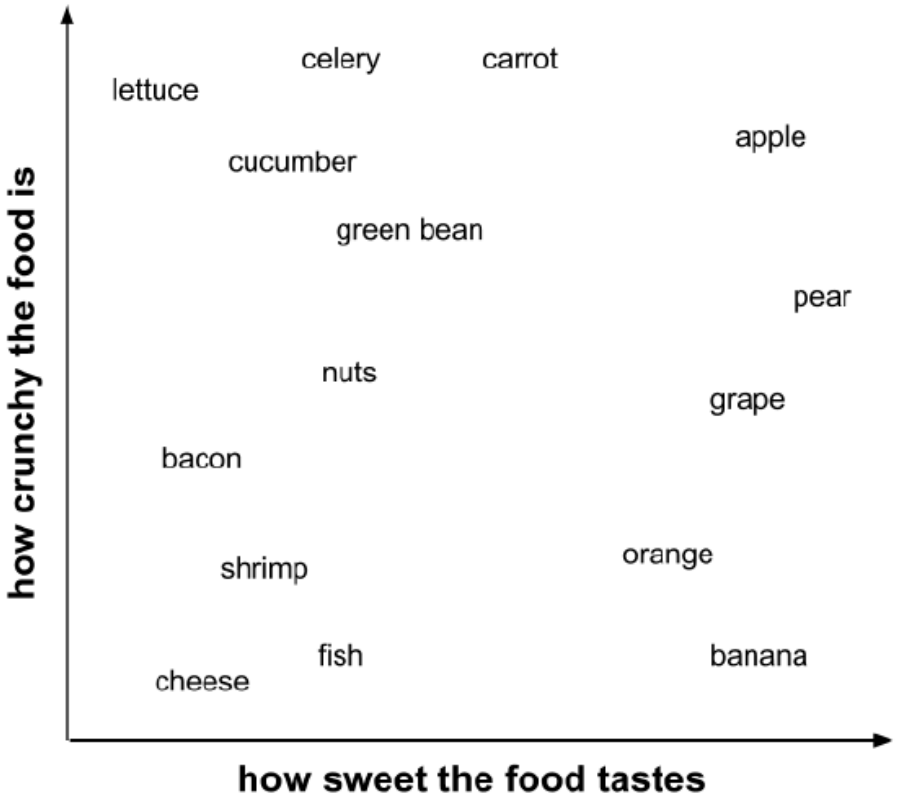
```
8
```

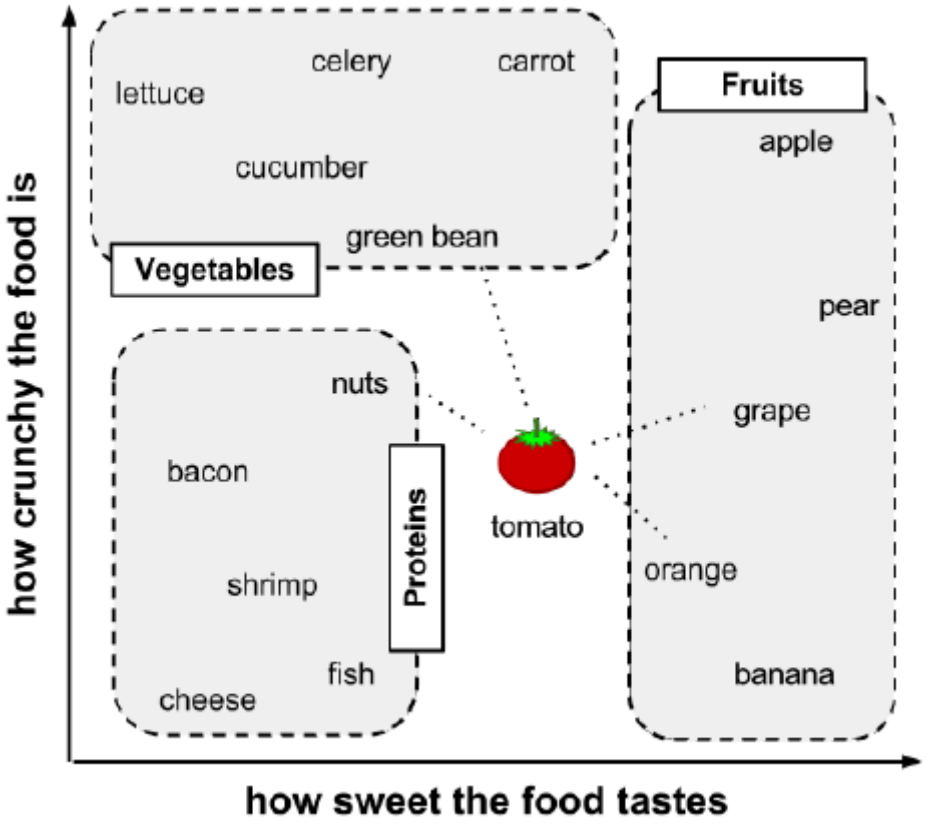
확장

거리개념과 kNN

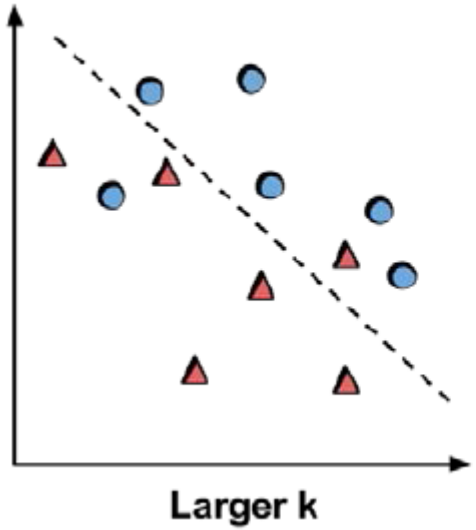
- kNN?
  - = 속성 데이터 간의 유사도를 기준으로 대상객체가 어떤 class에 속하는지 예측하는 알고리즘
  - 예: Blind testing을 통한 tomato 분류배정

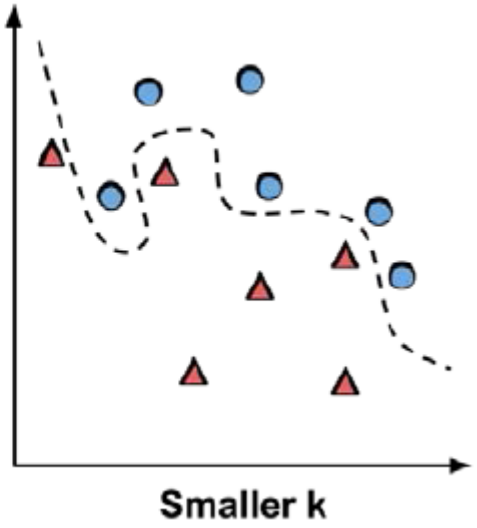
ingredient	sweetness	crunchiness	food type
apple	10	9	fruit
bacon	1	4	protein
banana	10	1	fruit
carrot	7	10	vegetable
celery	3	10	vegetable
cheese	1	1	protein


how crunchy the food is
how sweet the food tastes
celery
carrot
lettuce
apple
cucumber
green bean
pear
nuts
grape
bacon
orange
shrimp
fish
banana
cheese


how crunchy the food is
how sweet the food tastes
celery
carrot
lettuce
cucumber
green bean
Vegetables
Fruits
apple
pear
grape
orange
banana
nuts
bacon
shrimp
fish
cheese
Proteins
tomato

ingredient	sweetness	crunchiness	food type	distance to the tomato
grape	8	5	fruit	*sqrt((6 - 8)^2 + (4 - 5)^2) = 2.2*
green bean	3	7	vegetable	*sqrt((6 - 3)^2 + (4 - 7)^2) = 4.2*
nuts	3	6	protein	*sqrt((6 - 3)^2 + (4 - 6)^2) = 3.6*
orange	7	3	fruit	*sqrt((6 - 7)^2 + (4 - 3)^2) = 1.4*





실습과제

- 붓꽃에는 setosa, versicolor, virginica 의 3가지 품종이 있다. 이제 다음의 4가지 속성데이터를 이용하여 품종을 알아맞추는 모델을 작성하세요.
- 4가지 속성 데이터는 다음과 같습니다.
  - 꽃 밭침 넓이 'sepal width (cm)’ 꽃 밭침의 길이 'sepal length (cm)'
  - 꽃 잎 넓이 'petal width (cm)’ 꽃 잎의 길이 'petal length (cm)'

- 입력: 훈련데이터: iris.csv 파일
- 출력: ((변형 가능))
  - 생성된 모델의 훈련데이터 기준 정확도를 표시.
  - 단, kNN의 k값을 선택 가능하게 할 것

- 참고 1: kNN이란 속성데이터를 기준으로 유클리드 거리가 작을 수록 유사도가 크다는 것을 이용하여 예측하는 알고리즘을 말한다.
- 참고 2: 유클리드 거리란

추천 (recommender) 알고리즘

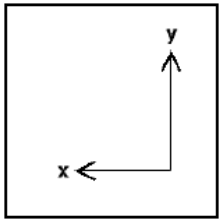
- 동작원리
  - User-Product 관계
  - Product-Product 관계
  - User-User 관계
- 데이터
  - 이용 데이터
    - 고객 Behavior : 예: Ratings, click 및 구매 이력
    - 고객 Demographic 데이터: 개인정보 관련 (예: 나이, 전공, 교육/소득, …)
    - 상품 속성 데이터 : 영화의 장르, 출연진, …
  - 데이터 수집
    - 평점 데이터: Explicit & Implicit (behavior: 클릭, view, 구매).
    - 상품 유사도 (Item-Item Filtering)
    - 고객 profile 유사도 (User-User Filtering), 단, cold start 문제

- 유사도 척도
  - Minkowski 거리

$$d(x,y) = (\sum_{i}^{n}(|x_i - y_i|)^q)^{\frac{1}{q}}$$

    - A generic distance metric
  - Manhattan 거리

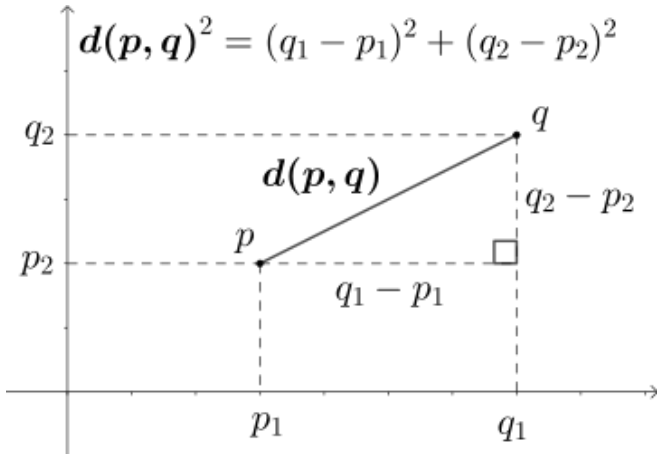
$$d(x,y) = \sum_{i}^{n}|x_i - y_i|$$



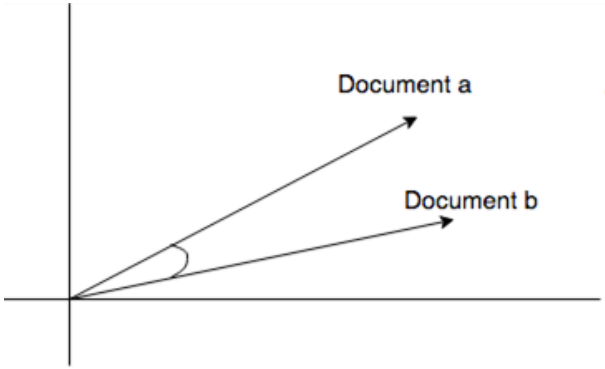
    - L1-distance/L1-norm
  - Euclidean 거리

$$d(x,y) = \sqrt{\sum_{i}^{n}(x_i - y_i)^2}$$

    - L2 norm



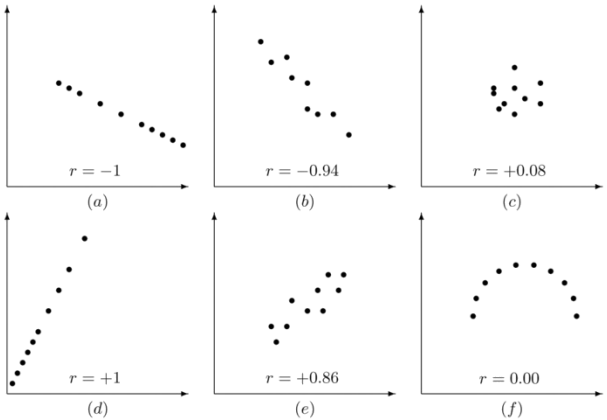
- Cosine Similarity



$$sim(a,b) = cos\theta = \frac{\vec{a}.\vec{b}}{\| \vec{a} \| \| \vec{b} \|}$$

- Pearson 상관계수

$$r = \frac{\sum (x-\bar{x})(y-\bar{y})}{\sqrt{\sum (x-\bar{x})^2 \sum (y-\bar{y})^2}}$$



- 기타
  - Jaccard Similarity, Hamming Distance, …

과제 실습

Chapter 19 2020년 삼성전자 기출문제

- Q 46 아기 상어
- Q 47 청소년 상어
- Q 48 어른 상어

실습과제

Q 46 아기 상어

난이도 ●●○ | 풀이 시간 50분 | 시간 제한 2초 | 메모리 제한 512MB | 기출 삼성전자 공채
링크 https://www.acmicpc.net/problem/16236

N × N 크기의 공간에 물고기 M마리와 아기 상어 1마리가 있습니다. 공간은 1 × 1 크기의 정사각형으로 나누어져 있습니다. 한 칸에는 물고기가 최대 1마리 존재합니다. 아기 상어와 물고기는 모두 크기를 가지고 있고, 이 크기는 자연수입니다. 가장 처음에 아기 상어의 크기는 2이고, 아기 상어는 1초에 상하좌우로 인접한 한 칸씩 이동합니다.

아기 상어는 자신의 크기보다 큰 물고기가 있는 칸은 지나갈 수 없고, 나머지 칸은 모두 지나갈 수 있습니다. 아기 상어는 자신의 크기보다 작은 물고기만 먹을 수 있습니다. 따라서 크기가 같은 물고기는 먹을 수 없지만, 그 물고기가 있는 칸은 지나갈 수 있습니다. 아기 상어가 어디로 이동할지 결정하는 방법은 아래와 같습니다.

- 더 이상 먹을 수 있는 물고기가 공간에 없다면 아기 상어는 엄마 상어에게 도움을 요청합니다.
- 먹을 수 있는 물고기가 1마리라면, 그 물고기를 먹으러 갑니다.
- 먹을 수 있는 물고기가 1마리보다 많다면, 거리가 가장 가까운 물고기를 먹으러 갑니다.
  - 거리는 아기 상어가 있는 칸에서 물고기가 있는 칸으로 이동할 때, 지나야 하는 칸의 개수의 최솟값입니다.
  - 거리가 가까운 물고기가 많다면, 가장 위에 있는 물고기, 그러한 물고기가 여러 마리라면, 가장 왼쪽에 있는 물고기를 먹습니다.

아기 상어의 이동은 1초 걸리고, 물고기를 먹는 데 걸리는 시간은 없다고 가정합니다. 즉, 아기 상어가 먹을 수 있는 물고기가 있는 칸으로 이동했다면, 이동과 동시에 물고기를 먹습니다. 물고기를 먹으면, 그 칸은 빈칸이 됩니다.

아기 상어는 자신의 크기와 같은 수의 물고기를 먹을 때마다 크기가 1 증가하는데 예를 들어, 크기가 2인 아기 상어는 물고기를 2마리 먹으면 크기가 3이 됩니다.

공간의 상태가 주어졌을 때, 아기 상어가 몇 초 동안 엄마 상어에게 도움을 요청하지 않고 물고기를 잡아먹을 수 있는지 구하는 프로그램을 작성하세요.

입력 조건

- 첫째 줄에 공간의 크기 N(2 ≤ N ≤ 20)이 주어집니다.
- 둘째 줄부터 N개의 줄에 공간의 상태가 주어집니다. 공간의 상태는 0, 1, 2, 3, 4, 5, 6, 9로 이루어져 있고, 아래와 같은 의미를 가집니다.
  - 0: 빈칸
  - 1, 2, 3, 4, 5, 6: 칸에 있는 물고기의 크기
  - 9: 아기 상어의 위치
- 아기 상어는 공간에 1마리 있습니다.

출력 조건

- 첫째 줄에 아기 상어가 엄마 상어에게 도움을 요청하지 않고 물고기를 잡아먹을 수 있는 시간을 출력합니다.

입력 예시 1

```
3
0 0 0
0 0 0
0 9 0
```

출력 예시 1

```
0
```

입력 예시 2

```
3
0 0 1
0 0 0
0 9 0
```

출력 예시 2

```
3
```

입력 예시 3

```
4
4 3 2 1
0 0 0 0
0 0 9 0
1 2 3 4
```

출력 예시 3

```
14
```

WRAP-UP

	1-13 (목)	1-14 (금)	1-17 (월)	1-18 (화)	1-19 (수)
오전 **(2Hrs)**	■ 강의 소개: ■ 코딩테스트 최근 동향과 학습방법 ■ 도 구 / 언 어 Review (OOP, standard lib.)	■ 주제(3): 탐욕알고리즘 ➢ Knapsack, … Huffman, … ➢ 예제실습 ➢ 과제실습	■ 주제(5): 탐색 ➢ 이진탐색, ➢ 선형… ➢ 예제실습 ➢ 과제실습	■ 주제(8): 동적프로그래밍 ➢ Memorization ➢ 예제실습 ➢ 과제실습	■ 주제(10): Graph(3) ➢ 기본형 review ➢ MSP (Kruskal, Bellman Ford, …) ➢ 예제실습 ➢ 과제실습
오후 (3Hrs)	■ 데이터구조 기초 ➢ :스택/큐 .. ■ 주제(1): 알고리즘 개요와 복잡도 ➢ time, space,.. ■ 주제(2): 솔루션 전략과 완전탐색 ➢ 솔루션 전략 ➢ 완전탐색	■ 주제(4): 정렬 ➢ 이진정렬, … ➢ 예제실습 ➢ 과제실습	■ 주제 6):Graph (1) ➢ DFS/BFS, … ➢ 예제실습 ➢ 과제실습 ■ 주제(7): 동적프로그래밍(1) ➢ Substructure/ subproblem, ➢ 예제실습 ➢ 과제실습	■ 주제(9) Graph (2) ➢ 그래프 일반론 ➢ 최단경로 (Dijkstra, …) ➢ 예제실습 ➢ 과제실습	■ 개발형 코딩 ➢ 구현 ➢ Web/DB ➢ 데이터과학 등 ■ 확장 ➢ NP-Complete, Approximation, .. ■ 마무리 ➢ 과정 review


Thank
you