

그리디

- 정의 - 현재 상황에서 지금 당장 좋은 것만 고르는 방법
- 대표적인 것은 다익스트라 알고리즘
- "그리디"인것을 확인하는 방법은 가장 큰 순서대로 or 가장 작은 순서대로와 같은 기준을 제시
- 주로 정렬 알고리즘과 짝은 이뤄서 출제
- 그리디는 두 가지 속성을 가지는 문제에서 유용하게 사용
- **Greedy Choice Property : 탐욕적 선택 조건**
한 번의 선택이 다음 선택과는 전혀 무관한 값
ex) 강의실 배정 문제, 거스름돈 문제
- **Optimal Substructure : 최적 부분 구조**
매 순간의 최적해가 문제에 대한 최적해여야 한다
ex) 최소 신장 트리(크루스칼), 다익스트라

탐욕적 선택 조건

1) 강의실 배정 문제

- 백준 문제 : <https://www.acmicpc.net/problem/1931>
- 최대한 많은 강의를 배치하는 문제
- 이 문제의 핵심은 강의 종료시간이 가장 빠른 수업 시간을 먼저 배정하는것
- 강의 시간이 가장 빨리 종료되는 최적해를 골랐을때, 이후 선택에서 최적해 고르는데 영향을 미치지 않는다.

강의 A AM 10:00 ~ PM 01:00

강의 B AM 10:00 ~ AM 11:00

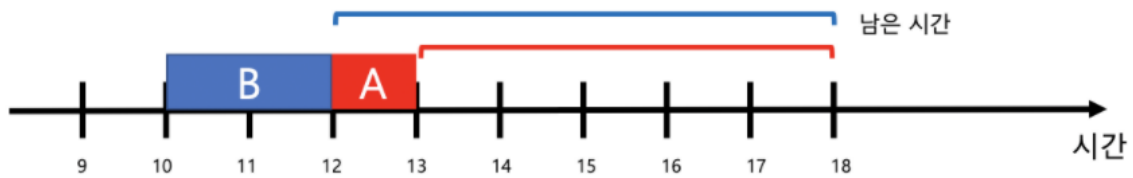


그림 1. Activity selection problem

2) 거스름돈

거스름돈 500, 100, 50, 10원 짜리가 무한히 존재, 손님에게 거슬러 줘야 할 돈이 **N원일때** 거슬러 줘야 할 동전의 최소 개수, 이때 N은 항상 10의 배수

해결방법

1. 가장 큰 화폐 단위부터 돈을 거슬러 준다.
2. 이유는 각 동전이 항상 작은 단위의 배수 이기 때문에 작은 단위의 동전들을 종합해 다른 해가 나올 수 없기 때문

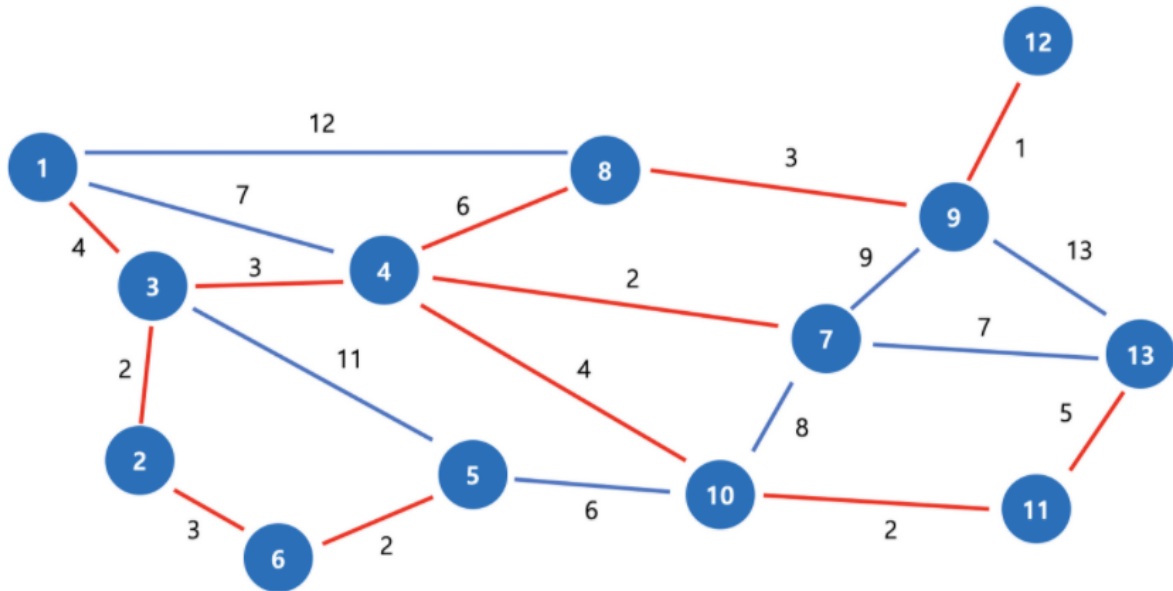
최적 부분 구조

1) 크루스칼, 최소신장트리

- 간선의 합이 최소가 되는 트리를 만드는 것이 목표
- 즉 모두 연결되어 있는 트리를 만든다 (이때 최소 비용을 사용)
- 간선의 비용은 각 노드를 연결하는 비용
- 우선순위 큐를 이용해 간선 생성 비용을 오름차순으로 정렬하고 각 노드를 연결하다가 트리가 되는 순간 완료
- 최단 경로를 찾는 것이 아닌 가장 적은 비용의 트리를 만들기 위해서 사용

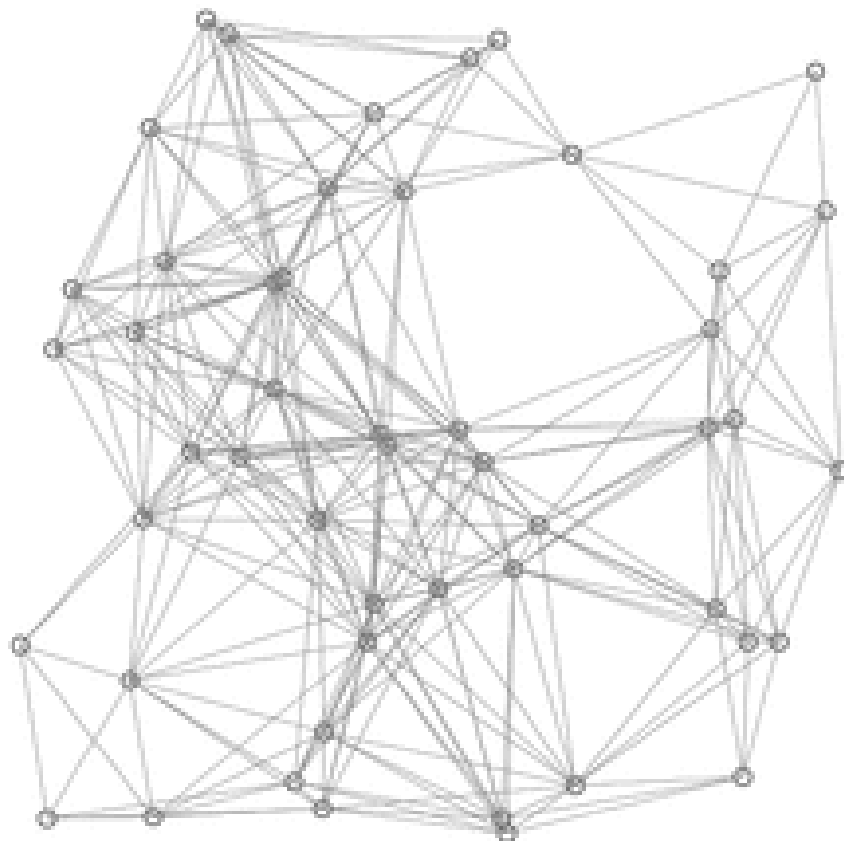
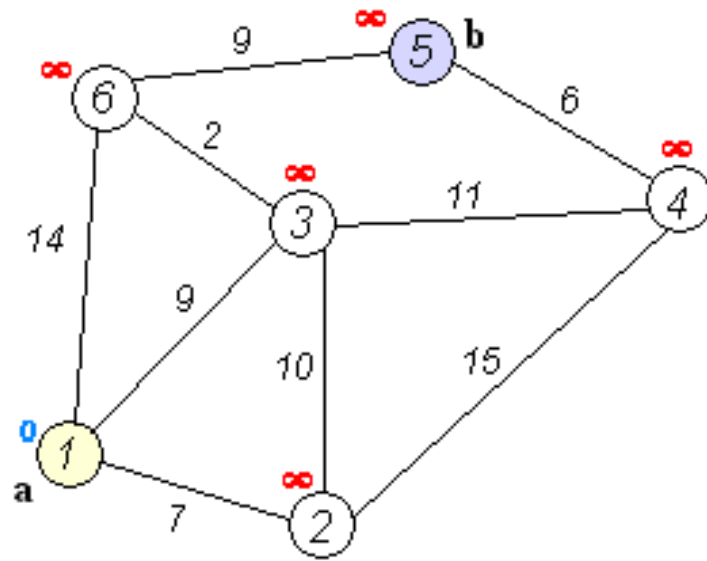
- 크루스칼 알고리즘은 서로소 집합을 만드는 (Union-Find)의 개념과 사이클의 개념이 필요

서로소 집합



2) 다익스트라

- 시작 정점부터 끝 정점까지 **가장 빨리 도달할 수 있는 경로를 찾는 알고리즘**
- 매 순간 최소 비용을 선택해 최종적으로 최단 경로를 찾아냄
- Dijkstra Algorithm (다익스트라 알고리즘)은 아래 그림과 같이 방향성을 가진 가중 그래프 $G=(V, E, w)$ (V : 정점의 집합, E : 간선의 집합, w : 간선의 가중치의 집합)와 시작점 s 가 주어졌을 때, **시작점으로부터 그래프의 임의의 정점까지 최단거리를 구해주는 알고리즘**이다.



서로소 집합의 사본

그리디 문제 풀이