

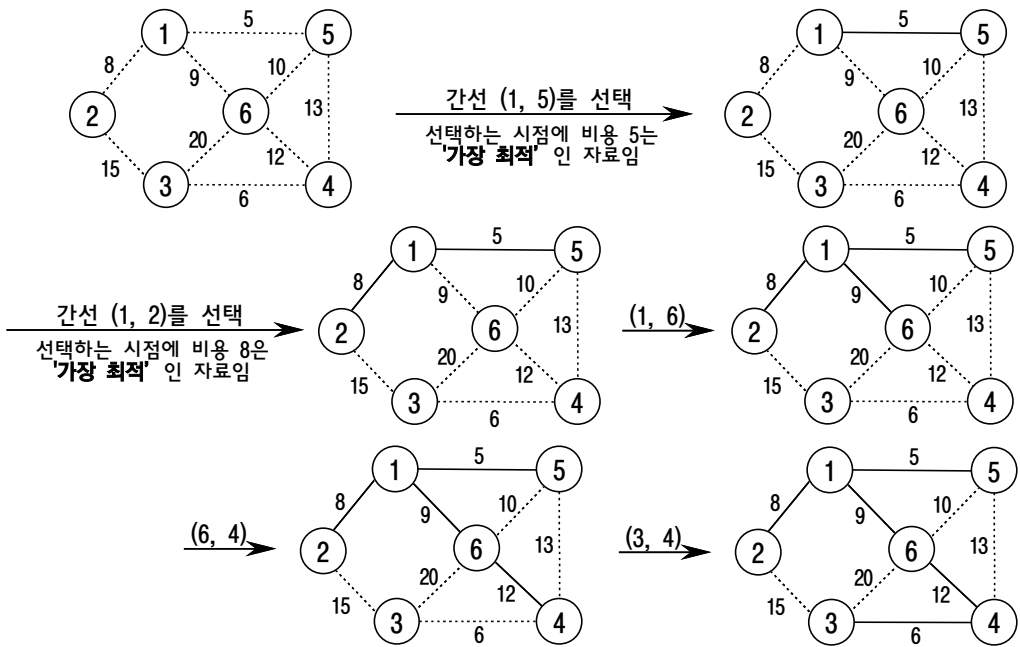
## 탐욕적인 방법(greedy method; 갈방법)

차례로 데이터를 선택하는데 이미 결정했거나 나중에 결정할 선택과는 무관하게 주어진 조건에 따라 선택할 당시에 '가장 최고'라고 생각되는 것을 항상 선택하는 방식이다.

- ① 탐욕적인 방법은 선택할 당시에 가장 최적인 것을 선택한다.
  - 각각의 선택은 부분적(locally)으로는 최적이 된다.
  - 전체적으로도 최적의 해답을 얻는 것이 희망이다. 하지만 항상 최적이 되지 않는다.
  - 따라서, 이 방법은 **항상 최적의 방법**이 되는지 **검사**해야 한다.
- ② 탐욕적인 방법은 사례를 더 작은 사례로 분할하지 않는다.
- ③ 탐욕적인 방법은 동적계획법처럼 최적화 문제 해결에 사용된다.
- ④ 예 : 최소비용 신장트리, 최단경로, 스케줄 작성, 배낭 채우기

**예제 1** 최소비용 신장트리 구하기 - Prim 알고리즘
















































다음은 Prim 알고리즘에 의한 최소비용 신장트리가 생성되는 과정을 나타낸 것이다.






- Prim 알고리즘은 탐욕적인 방법이다.
- Prim 알고리즘은 선택할 당시에 '가장 최고'라고 생각되는 것을 항상 선택하는 방식이다.

예제 2 거스름돈을 지불하는 경우

조건 : 거스름돈은 정확하게 지불하면서 동전 수를 가능한 적게 지급해야 한다.

[사례 1] 거슬러 줄 금액 : 71원 동전 종류 및 개수	[사례 2] 거슬러 줄 금액 : 16원 동전 종류 및 개수
    2개    5개    5개    5개	    1개    1개    1개    5개
① 50원  ② 50원  <del></del> ③ 10원   ④ 10원    ⑤ 10원    <del></del> ⑥ 5원    <del></del> ⑦ 1원     (최적의 해가 된다)	① 12원  ② 10원  <del></del> ③ 5원  <del></del> ④ 1원   ⑤ 1원    ⑥ 1원     ⑦ 1원      (최적의 해가 아니다)

- 사례 2에서 최적의 해는    으로 동전 수는 3개이어야 한다.
- 이는 탐욕적인 방법이 항상 최적의 해를 구해주는 것은 아니라는 것을 보여준다.

**Tip** 우리나라의 동전구조는 탐욕적인 방법으로 거스름돈을 지불하면 항상 최적의 해가 되는 것으로 증명되어 있다.(단, 준비된 동전 수가 부족하지 않을 경우이다)

**기출문제 분석**

1. 다음에서 설명하는 알고리즘 설계 기법으로 가장 알맞은 것은? [2018년 계리직]

해결하고자 하는 문제의 최적해(optimal solution)가 부분 문제들의 최적해들로 구성되어 있을 경우, 이를 이용하여 문제의 최적해를 구하는 기법이다.

- ① 동적계획법(dynamic programming)
- ② 탐욕적 알고리즘(greedy algorithm)
- ③ 재귀 프로그래밍(recursive programming)
- ④ 근사 알고리즘(approximation algorithm)

☞ 동적계획법

- 동적계획법은 먼저 작은 사례를 해결하고 그 결과를 저장한다.  
→ 나중에 저장된 결과가 필요하면 이용한다.
- 동적계획법은 배열을 이용한 상향식 접근 방법이다.  
→ 동적계획법은 배열을 이용한 반복적 알고리즘을 일컫는다.
- 예 : 최단경로, 피보나치수 구하기, 이항계수 구하기, 최적 이진탐색트리, 외판원 문제

정답 : ①

2. 갈망법(greedy method) 알고리즘에 대한 설명이다. 옳지 않은 것은? [2002년 서울 7급]

- ① Huffman Tree는 갈망법 원리로 만들어진다.
- ② 운영체제의 잡스케줄러는 갈망법으로 구현할 수 있다.
- ③ 갈망법으로 항상 전체적으로 최적의 해를 구할 수 있다.
- ④ Kruscal Algorithm과 Prim Algorithm은 갈망법을 이용한 알고리즘이다.
- ⑤ 단계별로 최적의 해를 구하는 방식이다.

☞ 탐욕적인 방법(greedy method; 갈망법)

- 갈망법은 전체적으로 최적의 해답을 얻는 것이 희망일 뿐 항상 최적이지 않다.
- 따라서, 갈망법은 항상 최적의 방법이 되는지 검사해야 한다.

정답 : ③