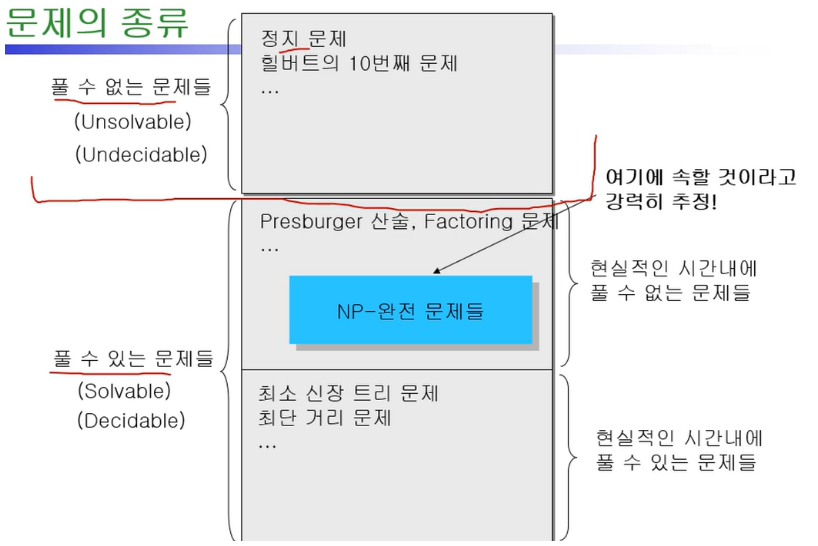
ch.07 NP-완전 문제

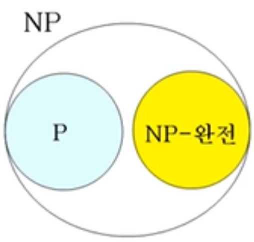
7.1 문제 분류

1. P(polynomial) 문제 집합 : O(n^k), 즉 다항식 시간 내에 해결 가능 ⇒ deterministic
2. 다항식보다 큰 시간복잡도를 가진 알고리즘으로 해결되는 문제 집합
    a. NP-완전 문제 집합 : exponential time의 시간복잡도 ⇒ nondeterministic
        i. 어느 하나의 NP-완전 문제에 대해서 다항식 시간의 알고리즘 존재 → 다른 것도



NP(Nondeterministic Polynomial) 문제 집합

Yes 대답이 나오는 해 제공 → Yes 대답을 내는 해라는 사실을 다항식 시간 내에 확인



- P 문제 집합 : 다항식 시간 내에 Yes, No 대답 가능

- NP - 완전 문제 집합 : 지수 시간 내에 Yes, No 대답 가능

찾아줘 냐를 묻는게 X

이게 이 문제의 해냐?로 변경

- NP 문제를 해결하기 위해서는 문제의 해가 yes, no가 되도록 주어진 문제 변형해야 함
    - 결정(decision) 문제로 변형 (decider)
    - 결정 문제 : yes, no가 되도록 주어진 문제

- ex) TSP 알고리즘
    - 각 도시를 1번만 방문하고 시작도시로 돌아오는 경로의 거리가 k보다 짧은 경로 있음?

– 8개 도시 (~~A B C D E F G~~ H)에 대한 여행자 문제의 NP 알고리즘은 다음과 같다. 단, A는 시작 도시이다.

– 8개 도시 (A B C D E F G H)의 여행자 문제의 하나의 해를 추측한다.
  - 예를 들어, A G D H F E B C를 추측했다고 가정

– 추측한 해의 값을 다음과 같이 계산한다.

해의 값 = (A와 G 사이의 거리)
+ (G와 D 사이의 거리)
+ (D와 H 사이의 거리)
……
+ (B와 C 사이의 거리)
+ (C와 A 사이의 거리)

– 그리고 해의 값이 K보다 작으면 'yes'라고 답한다.

7.2 NP-완전 문제의 특성

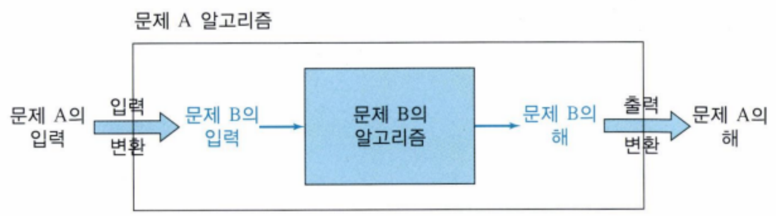
변환(reduction, 환원)

문제 A를 해결하기 위해서 문제 A의 입력 형태를 문제 B의 입력 형태로 변환

→ 변환된 입력으로 B 알고리즘 수행

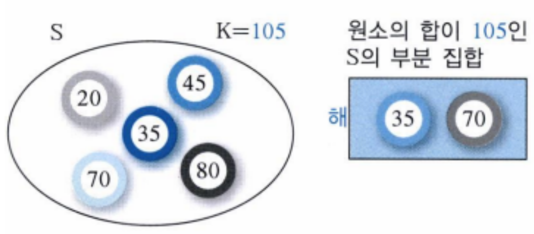
→ 수행 결과인 해를 문제 A의 해로 변환

→ 문제 A 해

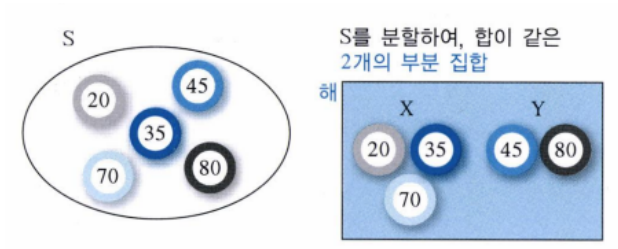


example A → B 환원!

- A : 부분집합의 합(subset sum)



- B : ‘동일한 크기의’ 분할(partition) 문제



→ X = 20 + 35 + 70 = 125 / Y = 45 + 80 = 125

1. A의 입력인 집합 S → B의 입력으로 변환
    a. t = s(집합 S의 모든 원소의 합) - 2K(A에 주어진 K값)
    b. B의 입력 S’ = S U {t}
2. B의 해인 X, Y의 원소 합 = (s-K) / t = S-2K
    a. S’의 모든 원소의 합 = s + t = s + s - 2K = 2s - 2K = 2(s - K)
3. A의 해 : B의 해 중 t를 가진 집합에서 t를 제거한 집합
    a. 만약 X에 t가 속해 있었다면, A의 해 : X - {t}
        i. A의 해인 원소의 합 : (s-K) - t = (s-K) - (s - 2K) = K

→ t

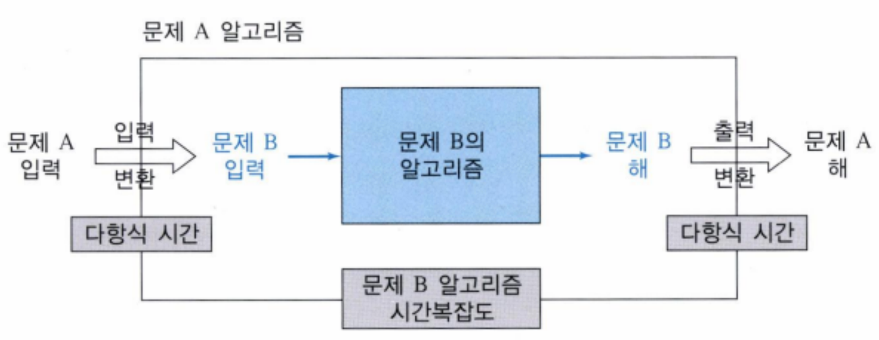
→ 추가해줬었으니 다시 빼면 됨!

time complexity

1. A의 입력을 B로 변환하는 시간
2. B를 위한 알고리즘 수행 시간
3. B의 해를 A의 해로 변환하는 시간

→ 1+2+3

- 1,3 → 단순한 입출력이기에 다항식 시간 내에 수행
- 2에 따라 시간복잡도가 결정
    - 만약 B가 다항식 시간이 걸린다면 A도 다항식 시간 내에 해결됨 → NP-완전 문제 관계



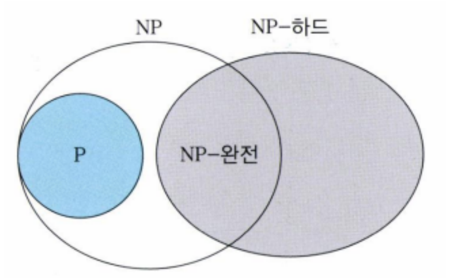
    - 이와 같은 관계가 성립 → polynomial time reduction
    - 만약 B가 C로 변환 가능 → A가 C로 다항식 시간에 변환 가능

⇒ 추이(transitive) 관계 : 어느 한 문제만 다항식 시간에 해결되면, 다른 NP-완전 문제들이 해결

NP-하드 문제

어느 문제 A에 대해서, 모든 NP 문제가 A로 다항식 시간에 변환이 가능하다면 → A는 NP-하드 문제

- 'hard' : 적어도 어떤 NP 문제보다는 어려움



→ A가 NP-완전 문제가 되려면, A = NP && A=NP-하드

7.3 NP-완전 문제의 소개

SAT(satisfiablility)

부울 변수들이 OR로 표현된 논리식이 여러 개 주어질 때,

이 논리식들을 모두 만족시키는 부울 변수 찾는 문제

[예제] 부울 변수 w, x, y, z에 대하여,

1) $(w \lor y)$, $(\overline{w} \lor x \lor z)$, $(\overline{x} \lor \overline{y} \lor \overline{z})$

해: w=true, x=true, y=false, z=true or false

2) $(w \lor \overline{x})$, $(x \lor \overline{y})$, $(y \lor \overline{w})$, $(w \lor x \lor y)$, $(\overline{w} \lor \overline{x} \lor \overline{y})$

해: 없음

부분집합의 합(Subset Sum) → 위에서 정리

분할(Partition) → 위에서 정리

0-1 배낭(knapsnack)

배낭에 담을 수 있는 물건의 합 중 최대 가치를 찾는 문제

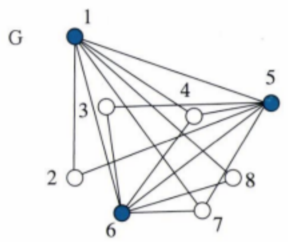
[예제] $C = 20kg$, $w_1 = 12kg$, $w_2 = 8kg$, $w_3 = 6kg$, $w_4 = 5kg$이고, $v_1 = 20$, $v_2 = 10$, $v_3 = 15$, $v_4 = 25$라면,

[해] 물건 2, 3, 4를 배낭에 담으면, 그 무게의 합은 8+6+5 = 19kg, 그 가치의 합은 10+15+25 = 50으로 최대가 된다.

정점 커버(Vertex cover)

그래프에서 간선의 양 끝점들 중에 적어도 1개의 점을 포함하는 집합

[예제]

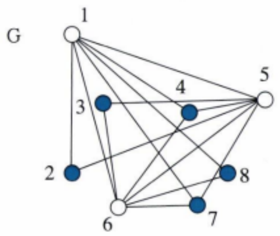


[해] {1, 5, 6}: 그래프의 각 간선의 양 끝점들 중에서 적어도 1개의 끝점이 점 1, 5, 6 중의 하나이다. 그리고 이는 최소 크기의 커버이다.

독립 집합(Independence Set)

그래프에서 서로 연결하는 간선이 없는 점들의 집합 중 최대 크기의 집합

[예제]

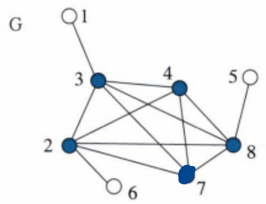


[해] {2, 3, 4, 7, 8}은 서로 간선으로 연결이 안 된 최대 크기의 독립 집합이다.

클리크(Clique)

그래프에서 모든 점들 사이를 연결하는 간선이 있는 부분 그래프 중 최대 크기

[예제]

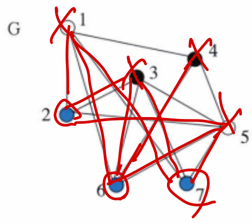


[해] {2, 3, 4, 7, 8}은 모두 간선으로 서로 연결된 최대 크기의 클리크이다.

그래프 색칠하기(Graph coloring)

주어진 그래프에서 인접한 점들을 서로 다른 색으로 색칠하는데 가장 적은 수의 색을 사용

[예제]



[해] {1, 5}는 흰색, {3, 4}는 검은색, {2, 6, 7}은 파란색으로 칠한다. 3가지 색보다 적은 수의 색으로 이 그래프를 칠할 수는 없다.

집합 커버(Set Cover)

주어진 집합 S = {1, 2, 3 .. n}에 대해서 S의 부분 집합들이 주어질 때, 부분 집합들 중에서 합집합하여 S와 같게 되는 부분 집합 → 집합 커버

집합 커버 중 가장 적은 수의 부분 집합으로 이루어진 집합을 선택

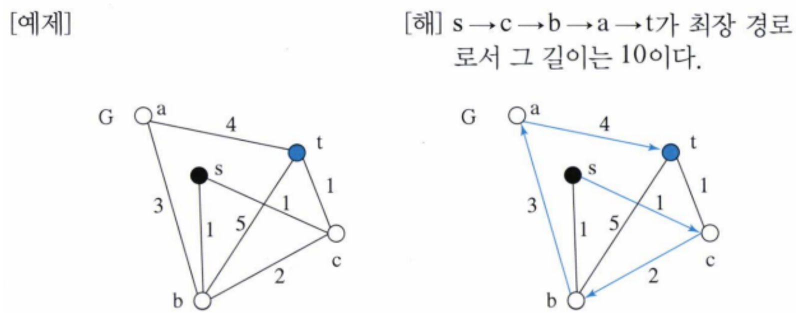
[예제] S = {1, 2, 3, 4, 5}, 부분 집합: {1, 2, 3}, {2, 3, 4}, {3, 5}, {3, 4, 5}라면,

[해] {1, 2, 3}과 {3, 4, 5}를 합집합하면 S가 되고, 부분 집합 수가 최소이다.

최장 경로(Longest Path)

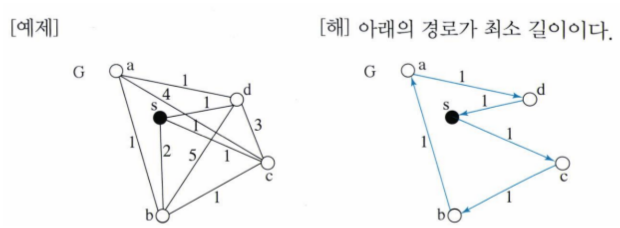
주어진 가중치 그래프 G=(V,E)에서 시작점 s에서 도착점 t까지의 가장 긴 경로를 찾는 문제

간선의 가중치 → 양수, 찾는 경로에는 반복되는 점이 없어야 함



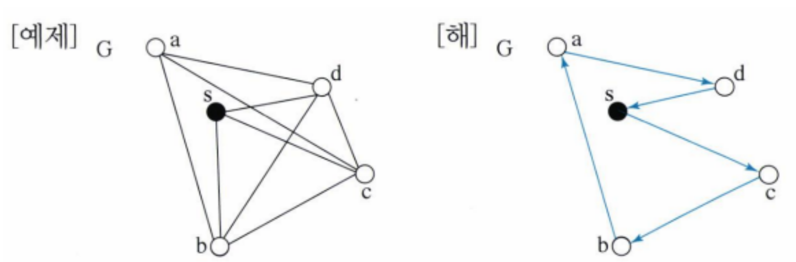
여행자(Traveling Salesman)

주어진 가중치 그래프 G=(V,E)에서 임의의 한 점에서 출발하여, 다른 모든 점들 1번씩만 방문 다시 시작점으로 돌아오는 경로 중에서 최단 경로를 찾는 문제



해밀토니안 사이클(Hamiltonian Cycle)

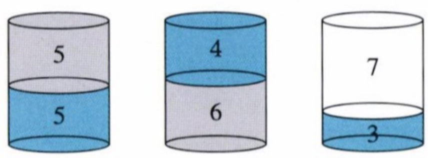
주어진 가중치 그래프 G=(V,E)에서 임의의 한 점에서 출발하여, 다른 모든 점들 1번씩만 방문 다시 시작점으로 돌아오는 경로 중에서 간선의 가중치를 모두 동일하게 하여 해를 찾는 문제



- **통 채우기**(Bin Packing): n개의 물건이 주어지고, 통(bin)의 용량이 C일 때, 가장 적은 수의 통을 사용하여 모든 물건을 통에 채우는 문제이다. 단, 각 물건의 크기는 C보다 크지 않다.

  [예제] 통의 용량 C=10이고, n=6개의 물건의 크기가 각각 5, 6, 3, 7, 5, 4 이면,

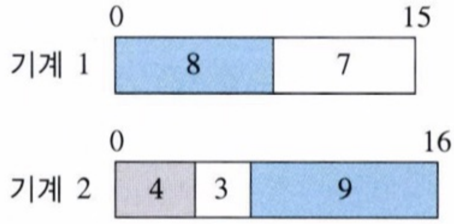
  [해] 3개의 통을 사용하여 다음과 같이 채울 수 있다.



- **작업 스케줄링(Job Scheduling):** n개의 작업, 각 작업의 수행 시간 t_i(단, $i = 1, 2, 3, \cdots, n$), 그리고 m개의 동일한 성능의 기계가 주어질 때, 모든 작업이 가장 빨리 종료되도록 작업을 기계에 배정하는 문제이다.

  [예제] $n = 5$개의 작업이 주어지고, 각각의 수행 시간이 8, 4, 3, 7, 9이며, $m = 2$대가 있다면,

  [해] 아래와 같이 작업을 배정하면 가장 빨리 모든 작업을 종료시킬 수 있다.



0-1 배낭 ⟷ 부분집합의 합
정점커버 ⟷ 독립집합
정점커버 ⟷ 집합커버
독립집합 ⟷ 클리크
그래프 색칠하기 ⟷ 클리크
→ NP 변환 다 외우자!!

① 배낭 문제의 입력 → 부분집합의 합 문제의 입력
부분집합의 합의 크기 = 배낭의 용량
주어진 집합의 원소 = 물건의 무게
$k = C$
$S = \{w_1, w_2, w_3, w_4 \cdots\}$

② 정점커버 : 간선의 양 끝점 중 하나만 포함 → 서로 연결될 수가 없음
독립집합 : 연결되어 있는 간선이 하나도 없음. → 가장 큰 부분집합
⇒ 정점커버에서 선택된 정점만 제외하면 독립집합의 해!

★③ 정점커버
집합커버 : 부분집합의 합이 S가 되는 경우 중 가장 적은 수의 부분집합
↳ 정점커버

④ 독립집합 : 연결되어있는 간선이 하나도 없음.
클리크 : 연결되어 있는 간선이 가장 큰 부분집합.
↳ 클리크에서 선택된 정점은 제외

⑤ 클리크에서 선택된 간선의 반대쪽 끝점은 다 다른 색으로 색칠.
① 클리크에서 선택된 간선이 더 이상 정점 선택할 때까지 같은 색,
② 선택된 정점만 남았을 때 색, change
→ 그 중 가장 많은 간선부터!
③ 선택되지 않은 간선이 없을 때까지 수행