

AI 알고리즘

정수계획법 개요

학습내용

- 배낭 문제 개요
- Knapsack 알고리즘

학습목표

- 배낭 문제의 정의와 분류에 대해 설명할 수 있다.
- Knapsack 알고리즘의 수행 과정에 대해 설명할 수 있다.

배낭 문제

배낭 문제 개요

배낭 문제 개요

❖ 배낭 문제

- n개의 물건과 각 물건 i의 무게 w_i 와 가치 v_i 가 주어지고, 배낭의 용량이 C일 때,
배낭에 담을 수 있는 물건의 최대 가치는
 - 단, 배낭에 담은 물건의 무게의 합이 C를 초과하지 말아야 하고, 각 물건은 1개씩만 있음

❖ 배낭 문제 분류

- 물건을 쪼개는 방식
 - 가치가 큰 물건부터 담고 남은 무게만큼 물건을 쪼개는 방식
 - 그리디 알고리즘으로 해결 가능
- 물건을 쪼갤 수 없는 배낭 문제
 - Brute-Force
 - 모든 경우의 수 고려
 - 열거법
 - Greedy
 - 가격이 높은 보석, 혹은 (가격/무게) 값이 높은 보석부터 넣기
 - 동적계획법(Dynamic Programming)
 - Devide-and-Conquer 방법

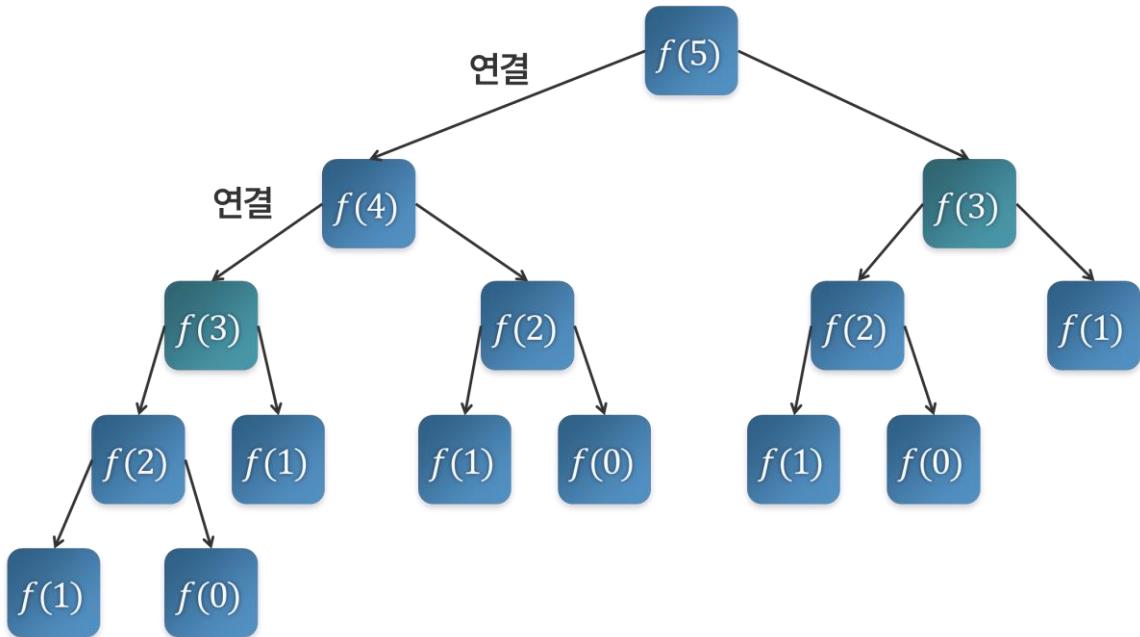
❖ Devide-and-Conquer 예시

- 피보나치 수열
 - 1, 1, 2, 3, 5, 8, 13, ...
 - $f(n)=f(n-1)+f(n-2)$

배낭 문제 개요

❖ Devide-and-Conquer 예시

- 피보나치 수열
 - 1, 1, 2, 3, 5, 8, 13, ...
 - $f(n) = f(n-1) + f(n-2)$



❖ 문제 살펴보기

- n 개의 물건과 각 물건 i 의 무게 w_i 와 가치 v_i 가 주어지고, 배낭의 용량이 C 일 때, 배낭에 담을 수 있는 물건의 최대 가치는?
 - 단, 배낭에 담은 물건의 무게의 합이 C 를 초과하지 말아야 하고, 각 물건은 1개씩만 있음

❖ 부분 문제

- 정의
 - $K[i, w]$
 - 물건 1~ i 까지만 고려하고, (임시) 배낭의 용량이 w 일 때의 최대 가치
 - 단, $i = 1, 2, \dots, n$ 이고, $w = 1, 2, \dots, C$
 - 문제의 최적해: $K[n, C]$

배낭 문제

Knapsack 알고리즘

Knapsack 알고리즘

❖ Knapsack 알고리즘

- 입력: 배낭 용량 C , n 개의 물건과 각 물건 i 의 무게 w_i 와 가치 v_i (단, $i=1, 2, \dots, n$)
- 출력: $K[n, C]$

```

01  for i = 0 to n    K[i,0]=0      //배낭의 용량이 0일 때
02  for w = 0 to C   K[0,w]=0      //물건 0이란 어떤 물건도 고려하지 않을 때
03  for i = 1 to n
04    for w = 1 to C           //w는 배낭의 (임시) 용량
05      if (w > w)           //물건 i의 무게가 임시 배낭 용량을 초과할 경우
06          K[i,w] = K[i-1, w]
07      else                  //물건 i를 배낭에 담지 않을 경우와 담을 경우 고려
08          K[i,w] = max{K[i-1, w], K[i-1, w - w_i] + v_i}
09  return K[n,C]

```

- 예제
 - 물건 i 를 가지고 가고 싶음
 - 배낭의 최대 허용량 w 를 구해 놓음
 - 배낭에서 물건 i 를 담을 공간 마련, $K[i-1, w - w_i]$
 - 물건 i 를 가지고 가고 싶음
 - 물건 1~(i-1)까지 고려하여 현재 배낭 용량이 w 인 경우 최대 가치



❖ Knapsack 알고리즘 수식

$$K[i, w] = \begin{cases} K[i-1, w] & \text{if } w_i > w \\ \max\{v_i + K[i-1, w - w_i], K[i-1, w]\} & \text{else} \end{cases}$$

Knapsack 알고리즘

❖ 수행과정

- 배낭의 용량 $C=10\text{kg}$
- 물건 4가지 종류

물건	1	2	3	4
무게	5	3	6	4
가치	10	60	20	40

- 2차원 배열 만들기

- ① $w = 1, 2, 3, 4$ 일 때, 각각 물건 1을 담을 수 없음
- ② $w = 5$ 일 때, $K[1, 5] = 10$
- ③ $w = 6, 7, 8, 9, 10$ 일 때, $K[1, 6] = K[1, 7] = K[1, 8] = K[1, 9] = K[1, 10]$
- ④ $w = 1, 2$ 일 때, $K[2, 1] = K[2, 2] = 0$
- ⑤ $w = 3, 4$ 일 때, $K[2, 3] = K[2, 4] = 60$
- ⑥ $w=5$ 일 때, $K[2, 5] = \max\{K[1, 5], K[1, 5] + v_1\} = \max\{10, 0+60\} = 60$
- ⑦ $w=6, 7$ 의 두 경우 모두 물건 1을 빼내고 물건 2를 배낭에 담는 것이 더 큰 가치
- ⑧ $w=8$ 일 때, $K[2, 8] = \max\{K[1, 7], K[1, 7] + v_1\} = \max\{10, 10+60\} = 70$
- ⑨ 이 후의 과정
- ⑩ 결과

배낭 용량 $w=$			0	1	2	3	4	5	6	7	8	9	10
무게	가치	물건	0	0	0	0	0	0	0	0	0	0	0
5	10	1	0	0	0	0	0	10	10	10	10	10	10
3	60	2	0	0	0	60	60	60	60	60	70	70	70
6	20	3	0	0	0	60	60	60	60	60	70	80	80
4	40	4	0	0	0	60	60	60	60	100	100	100	100