

# 과제 4: 0-1Knapsack Problem(Backtracking)

20161024 백우열

## 1. 구현:

- 구현 언어: C++

### 1) Knapsack(Algorithm 5.7):

- 파일:
  - egData: example 5.6
    - 구현: knapsack.hpp
    - 출력: main.cpp
  - myData: 자작 데이터
    - 구현: knapsack.hpp
    - 출력: main.cpp
- Problem: 상태공간트리를 이용해 backtracking depth-first search 방식으로 knapsack problem 해결
- Inputs:
  - $n$  = items 개수
  - $W$  = weight 한계
  - $w[]$  = item 의 weight 배열( $p[i]/w[i]$ 의 내림차순으로 정렬됨)
  - $p[]$  = item 의 profit 배열( $p[i]/w[i]$ 의 내림차순으로 정렬됨)
- Outputs:
  - bestset[] = item 별 선택 여부(yes: 선택, no: 미선택)
  - maxprofit = 최대로 얻을 수 있는 profit

## 2. 소스코드 출력:

### Example 5.6

```
[Running] cd "/Users/wooyeolbaek/21-2Algo/Assign4/final/egData/" && g++ main.cpp -o main &&  
"/Users/wooyeolbaek/21-2Algo/Assign4/final/egData/"main  
maxprofit: 90  
bestset: yes,no,yes,  
[Done] exited with code=0 in 1.613 seconds
```

### 자작 데이터

```
[Running] cd "/Users/wooyeolbaek/21-2Algo/Assign4/final/myData/" && g++ main.cpp -o main &&  
"/Users/wooyeolbaek/21-2Algo/Assign4/final/myData/"main  
maxprofit: 78  
bestset: yes,yes,no,yes,  
[Done] exited with code=0 in 0.871 seconds
```

### 3. 자작 데이터 손계산:

