과목명: 알고리즘

담당 교수: 이창훈 교수님

학과: 컴퓨터공학과

학번: 19101407

성명: 자보히르

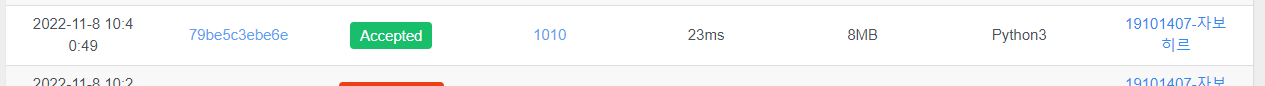
구분: 보고서

**배낭 문제 (Knapsack Problem) (Greedy &DP)**

Diagram, text

Description automatically generated

해결: 1\_DP 알고리즘, 79be5c3ebe6e



코드의 이 부분에서 표시된 것처럼 :

Text

Description automatically generated

물체의 무게에 추가적인 제한이 있을 때, 배낭 문제는 동적 프로그래밍 방법으로 의사 다항식 시간 내에 해결될 수 있습니다. 즉,

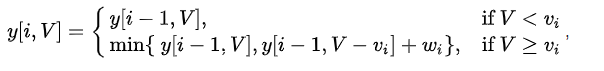
Chart, diagram

Description automatically generated with medium confidence

m[n,W]를 계산하면 정확한 솔루션을 찾을 수 있습니다. m[i,w] 배열이 기계 메모리에 있는 경우, 이 알고리즘이 가장 효율적인 알고리즘 중 하나일 가능성이 높습니다.

우리는 DP 알고리즘에서 볼 수 있는 것처럼 재귀적 비율은 물체의 무게 뿐만 아니라 가치에 대해서도 기록할 수 있습니다. 이를 위해 첫 번째 i 항목에서 얻을 수 있는 총 가치 V의 최소 항목 중량을 y[i, V]로 표시합니다. 필요한 중량을 얻을 수 없는 경우 y[i, V]=W+1으로 표시할 수 있습니다.

그런 다음 동적 프로그래밍의 함수 방정식을 해결합니다. 즉:



그리고 다음에의 시작 조건 포함도 있어야 되고:

Diagram

Description automatically generated

2\_GREEDY 알고리즘: dc671a88c351

2\_GREEDY 실습 때 같은 문제를 greedy 알고리즘으로 해결하기 위해 여러 번 노력해봤지만 오류가 계속 나오고 있었습니다. 실습 때 예제 코드를 보고 다시 실행해봤지만 오류 다시 나왔습니다. :

Text

Description automatically generated

똑 같은 코드는 <https://www.onlinegdb.com/online_python_compiler> 에서 실행해 봤을 때 결과가 잘 나왔습니다:

A screenshot of a computer

Description automatically generated

알고 보니까 <http://203.246.84.127/problem> 에서 Greedy 알고리즘으로 배낭 문제를 해결해서 코드 실행 때 오류가 나오는 이유가 서버에서 배낭 문제의 DP 코드가 실행될 수 있지만 Greedy 알고리즘으로 되어있는 코드가 실행되지 않다는 것을 알게 됐습니다.

배낭 문제 를 Greedy 알고리즘으로 풀 떄: 이 알고리즘의 작동 시간은 정렬 시간과 배치 시간으로 구성됩니다. 정렬 복잡성은 O(N\log(N)입니다. 그러나 다음에 총 O(N) 시간 동안 배낭에 얼마나 많은 항목이 들어가는지 계산합니다. 필요한 경우의 최종 복잡성은 O(N\log(N)이고 이미 정렬된 데이터에서는 O(N)입니다.

Greedy알고리즘은 최적의 알고리즘과는 거리가 먼 답을 이끌어 낼 수 있다는 것을 알고 있어야 될까 봅니다. 그리고 동시에 무제한 배낭 문제에 대한 동일한 알고리즘은 최적 값의 최소 50%에 해당하는 응답을 제공하기도 합니다.

두 번째 Greedy 알고리즘으로 되어 있는 코드에서 볼 수 있는 것처럼 Greedy 알고리즘 기본 아이디어는 다음과 같습니다:

* benefit(가치) / weight(무게) 의 비율을 각 물품마다 구한다
* 비율이 높은 순으로 정렬을 한다
* 배낭에 물품들을 비율 높은 순으로 넣는다.
  + 만약에, 물품의 무게가 무게 최대치보다 크다면
    - (물품의 비율) \* (최대치까지 남은 무게)를 구해서 답에 더해준다
  + 아니라면,
    - 물품을 그대로 배낭에 넣고, 배낭에 넣을 수 있는 최대치 무게 값 들어간 무게만큼 줄인다.