"중등부 2번. 가방" 문제 풀이

작성자: 구재현

부분문제 1

백트래킹을 통해서, 가방에 담을 물건들의 모든 경우의 수를 열거한다. 각 경우 S에 대해서:

- 가방에 담기게 되는 물건의 무게 합 (A_S)
- 도둑이 남은 물건을 훔쳐갔을 때의 무게 합 (B_S)

를 계산할 수 있다. 모든 2^N개의 경우에 대해서 이를 배열에 저장해 두자. 각 x에 대한 문제의 정답은, $A_S \leq x$ 인 S 중 B_S의 최댓값이 된다. 시간 복잡도는 $O(2^N \cdot C)$ 이다.

부분문제 2

먼저 입력으로 주어진 모든 물건들을 무게 오름차순으로 정렬한다. 이 경우, 도둑은 내가 고르지 않은 물건들 중 맨 앞에 등장하는 최대 K개의 물건을 훔칠 것이다.

동적 계획법을 사용한다. $dp[i][j][w]$ = (첫 i 개의 물건에 대해서 가방에 넣을지 말지를 결정했으며, 도둑이 j 개의 물건을 들고 갔고, 지금까지 가방에 들어간 물건의 무게 합이 w 이하일 때, 도둑이 훔쳐가는 물건 무게의 최댓값) 으로 정의하면:

- $i+1$번째 물건을 가방에 넣는 경우: $dp[i+1][j][w+A_{i+1}]$ 로 상태 전이한다.
- $i+1$번째 물건을 가방에 넣지 않는 경우: 만약에 도둑이 K 개 미만의 물건을 훔쳐갔다면, $dp[i+1][j+1][w]$ 로 전이하며, A_{i+1} 만큼의 무게를 도둑이 훔쳐간다고 추가한다. 그렇지 않은 경우, $dp[i+1][j][w]$ 로 전이하며, 도둑이 추가로 훔쳐가는 무게는 없다.

각 x 에 대해, $i = N, 0 \leq j \leq K, w = x$ 인 경우 중 $dp[i][j][w]$ 의 최댓값을 계산하면 된다. 시간 복잡도는 $O(N^2 \cdot C)$ 이다.

부분문제 3

물건이 무게 순으로 정렬되었고 도둑은 가벼운 물건부터 훔쳐간다. 따라서, 어떠한 인덱스 $0 \leq i \leq N$에 대해, 도둑은 인덱스 i 이하의 물건 중 가방에 들어있지 않은 물건을 모두 훔쳐가고, 인덱스 i 초과의 물건을 훔치지 않는다. 이 인덱스 i 를 고정시켰을 때 도둑이 훔쳐가는 물건의 무게 합을 최대화하려면:

- $1, 2, \ldots, i$ 번 물건 중 정확히 $i - K$개의 물건을 가방에 담아야 한다.
- 가방에 담는 물건의 무게는 x 이하여야 하며, 가능한 작아야 한다.

만약에 위 조건을 만족하게끔 $w \leq x$ 만큼의 물건을 가방에 담을 수 있었다면, 도둑은 $\sum_{j=1}^{i} A_j - w$ 만큼을 훔쳐가게 된다.

정확히 $i - K$ 개의 물건을 담는다는 조건은 $i - K$개 이상의 물건을 담는다는 조건으로 바꿀 수 있다. 도둑이 인덱스 i 초과의 물건을 가져가는 경우도 고려되지만, 해당 경우는 실제 도둑이 훔쳐가게 될 무게 이하의 값이 고려되게 되어서 문제가 되지 않는다.

이제 동적 계획법을 사용한다. $dp[i][w]$ = (첫 i 개의 물건에 대해서 가방에 넣을지 말지를 결정했으며, 지금까지 가방에 들어간 물건의 무게 합이 w 이하일 때, 가방에 있는 물건의 개수 최대) 로 정의하자. 각 x 에 대해서, $0 \le i \le N, w = x$ 인 경우 중 $dp[i][w]$ 의 최댓값을 계산하면 된다. 시간 복잡도는 $O(N \log N + NC)$ 이다.

부분문제 4

도둑이 훔쳐가는 물건이 단 하나이다. 도둑이 $1 \le i \le N$번 물건을 훔쳐가게 하기 위해서는 $\sum_{1 \le j \le i-1} A_j$ 가 x 이하여야 한다. 이를 만족하는 최대 i를 $x = 1, 2, \ldots, C$ 에 대해 모두 구해주면 된다. x 가 증가함에 따라 최대 i도 증가하니, 현재까지의 최대 i 와 $\sum_{1 \le j \le i-1}$ 을 관리해 주면 (two pointers) 문제를 $O(N \log N + C)$ 에 해결할 수 있다.

부분문제 5

전체 문제는 그리디 알고리즘을 사용하여 해결할 수 있다. 다음 두 가지를 관찰하자:

- $N - K$ 개 초과의 물건을 가져갈 필요는 없다: 추가로 가져간 물건을 가방에서 빼면, 도둑이 해당 물건을 가져가게 된다.
- i번 물건을 가져가지 않고 $i + 1$번 물건을 가져갈 필요는 없다: $i + 1$번 물건 대신 i번 물건을 가방에 담으면, 가방의 무게는 그대로거나 감소하고, 도둑이 가져가는 물건의 무게는 그대로거나 증가한다. (i 번 물건이 들어갈 자리에 $i + 1$번 물건이 들어감)

두 조건에 의해서, 우리가 가져와야 할 물건은 가장 가중치가 낮은 i $(0 \le i \le N - K)$ 개의 물건임을 알 수 있다. 이러한 물건을 가져가면, 도둑은 $i + 1, i + 2, \ldots, i + K$번 물건을 가져가게 된다. 즉, 각 x에 대해 $\sum_{1 \le j \le i-1} A_j$가 x 이하인 최소 i 를 구하고, $i + 1, i + 2, \ldots, i + K$번 물건의 가중치 합을 출력하면 된다. 앞부분은 부분문제 4와 동일하고, 뒷 부분은 부분합을 사용하면 된다. 전체 문제를 $O(N \log N + C)$에 해결할 수 있다.