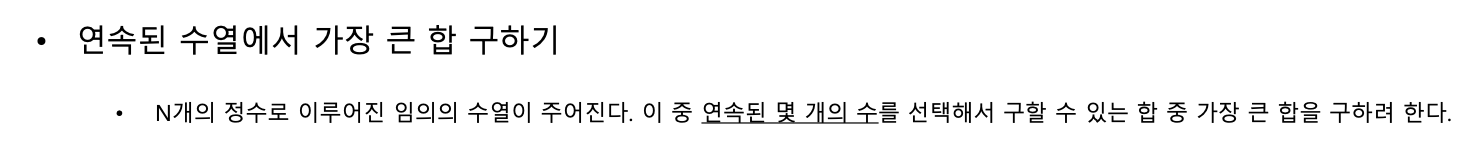
고급 소프트웨어 실습

분반: 1반(월요일)

학번: 20181662

이름: 이건영

1. 실습 3이 Greedy algorithm과 Dynamic programming 중 어디에 해당된다고 할 수 있는가? 이유를 설명하시오.



위의 문제는 Dynamic programming 방식에 더욱 적합하다고 할 수 있다. 어떤 문제를 여러 개의 부분 문제로 쪼갤 수 있고, 그 부분 문제에서 구한 최적의 답으로 원래 문제의 답을 구할 수 있다면 Dynamic programming 방식으로 문제를 풀 수 있는데, 위 문제 역시 그러한 경우이기 때문이다. 위 문제는 다음과 같은 작은 문제로 나눌 수 있다.

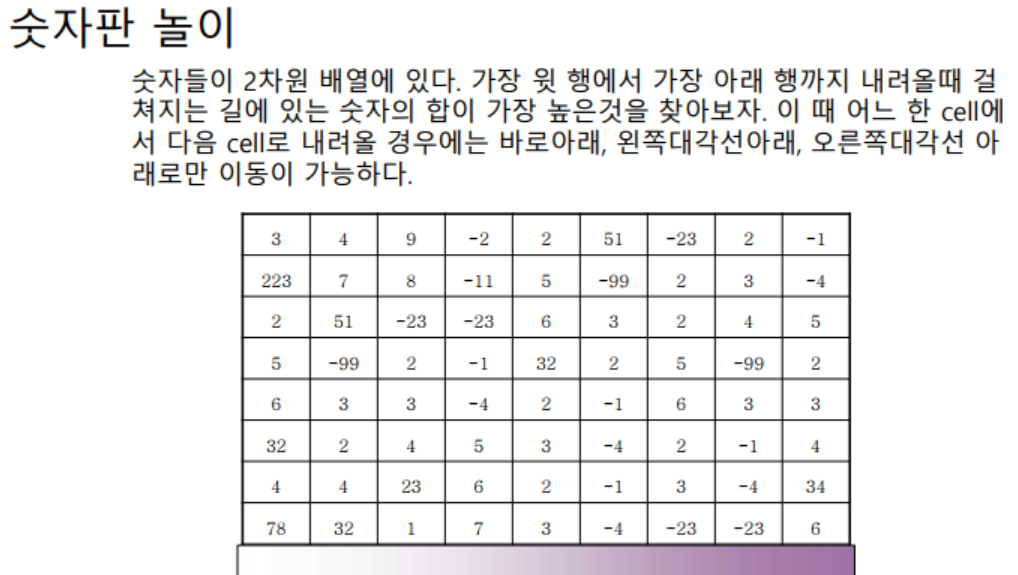
위의 방식으로 문제를 접근하여 해결 할 경우 O(n)의 시간복잡도로 문제를 해결할 수 있다.

*텍스트이(가) 표시된 사진

자동 생성된 설명*

2. 앞에서 다루지 않은 문제 중에 Dynamic programming을 사용했을 때 더 효율적으로 풀 수 있는 문제를 2개 들고, 알고리즘을 설명하시오. brute force 방식에 비하여 DP를 사용하는 경우 어느 정도 빨라지는지도 설명하시오.

1) 숫자판 놀이 문제



위 문제를 brute force 방식으로 푼다면, 한 위치에서 가능한 경우의 수가 3가지이므로 숫자판의 높이 n에 대하여 O(3^n)의 시간복잡도를 가지게 될 것이다. 그러나 각 칸에 대해 위의 세 경우의 수의 max값을 계산하여 중복되는 경우를 없앤다면 O(n^2)까지 시간복잡도를 줄일 수 있다. 식은 다음과 같다.

위 식을 통해 각 칸의 최댓값을 미리 계산할 수 있다.

2) 1로 만들기

텍스트이(가) 표시된 사진

자동 생성된 설명

위와 비슷하게 한 상황에서 가능한 동작이 3가지이므로 brute force 방식의 경우 O(3^N)의 시간복잡도를 가지게 된다. 그러나 위와 비슷하게 각 정수마다 도달할 수 있는 연산의 최소 횟수를 계산하는 방식으로 시간복잡도를 줄일 수 있다.

위의 방식처럼 중간 과정을 기억하는 Memoization방식을 사용하여 시간복잡도를 O(n)으로 줄일 수 있다.