

C5. Ski Trail

이원빈 백하현

Problem Abstraction

$n \times n$ grid 형식으로 수들이 주어질 때, 주어진 조건을 만족하는 모든 수열 (s_1, s_2, \dots, s_k)에 대하여 수열을 구성하는 모든 인접한 수들의 차이의 합의 최댓값 구하기

조건 1. s_i 와 s_{i+1} 은 $n \times n$ grid에서 서로 인접한 수이다.

조건 2. $s_i > s_{i+1}$ 이다.

단, $1 \leq n \leq 500$, $0 \leq s_i \leq 10^8$ 이다.

Forming Idea

- 수열 (s_2, \dots, s_k)는 (s_1, s_2, \dots, s_k)에 포함된다. 따라서 수열 겹치는 것을 방지하기 위해, $n \times n$ grid 중 가장 큰 수부터 내림차순으로 s_1 으로 설정하여 수열을 구한다.
 - $n \times n$ grid에 있는 임의의 수 x 에 대해, x 가 이미 어떠한 수열에 포함되었다면, (x, s_2, \dots, s_k)은 구하지 않아도 된다.
- (s_1, s_2, \dots, s_k)의 difficulty는 $(s_1 - s_2) + (s_2 - s_3) + \dots + (s_{k-1} - s_k) = s_1 - s_k$ 이므로, 결국 시작점과 끝점의 차이가 곧 해당 수열의 difficulty이다.
- 따라서 임의의 cell을 s_1 으로 하여 만들어질 수 있는 모든 수열의 difficulty 중 가장 큰 값(local_max)을 찾고, local_max들 중 가장 큰 값이 곧 구하고자 하는 값이다.

Definition

cell[i][j]: $n \times n$ grid에서 $i+1$ 번 행의 $j+1$ 번째 수

cell[i][j]_adj: cell[i][j]에 대하여 위, 아래, 오른쪽, 왼쪽으로 인접한 수

connected: cell[s_i][s_j] > cell[s_i][s_j]_adj일 때 connected 되었다고 정한다.

local_max: cell[i][j]으로 시작하는 모든 수열 (cell[i][j], s_2, \dots, s_k)에 대한 cell[i][j] - s_k 값의 최대값

bfs_cell(s_i, s_j): cell[s_i][s_j]를 시작점으로 하여 connected된 값들을 너비 우선 탐색하는 operation

너비 우선 탐색을 통해 모든 (cell[s_i][s_j], s_2, \dots, s_k)를 확인하여 각 수열의 s_k 중 가장 작은 값 min을 찾는다.

이후 cell[s_i][s_j] - min을 통하여 local_max를 구한다.

global_max: local_max의 최대값

Solution

- $n \times n$ grid의 수들을 입력받아 **cell**에 저장한다.
- cell[i][j]**를 기준으로 (cell[i][j], i, j)를 내림차순으로 정렬하여 **sorted_cell**에 저장한다.
- global_max**를 0으로 초기화 한다.
- sorted_cell[i]**에 대해 **bfs_cell**을 수행하여 **sorted_cell[i]**로 시작하는 수열들로부터 **local_max**를 얻는다.
- local_max > global_max**이면 **global_max**를 **local_max**로 갱신한다.
- 4~5를 $i = 0$ 에서 n^2-1 까지 반복한다.
- global_max**를 출력한다.

Time complexity

(2) $n \times n$ 개(n^2)의 cell을 sort $\rightarrow O(n^2 * \log(n^2)) = O(n^2 * \log n)$

(4) $n \times n$ grid의 각 cell에 있어서 최대 네 방향으로의 탐색이 가능하고, 이미 탐색한 cell은 다시 탐색하지 않으므로 4번의 탐색 x 모든 cell의 개수.
 $4 * O(n^2) = O(n^2)$

\rightarrow 따라서 time complexity는 $O(n^2 * \log n)$