

EX우주선착륙

24. 07. 06

김 태 현

문제 개요

- **행성 맵 정보가 1,000 * 1,000 크기로 주어진다.**
맵의 각 셀은 0~124 범위의 높이를 가진다.
- **우주선이 10,000대 주어진다.**
우주선은 가로(2~5), 세로(2~5) 크기 정보를 갖는다.
우주선의 네 모서리에 125~131 범위의 조절 가능한 다리가 있다.
우주선의 네 모서리가 위치한 맵의 높이 차가 6 이하면 착륙이 가능하다.
우주선은 90도 회전할 수 있다.
한 우주선이 착륙한 영역에 겹쳐서 착륙할 수 없다.
- **process(rows[], cols[], dirs[]) 함수에서 우주선을 착륙시켜 점수를 얻는다.**
착륙시킬 우주선 i의 좌상단 좌표와 회전 여부(1/0)를 rows[i], cols[i], dirs[i] 에 기록한다.
- 점수는 착륙시킨 모든 우주선의 가로(h), 세로(w) 크기로 다음과 같이 계산한다.
SCORE += h * w * min(h,w)
점수가 최대가 되게 우주선을 착륙시키자!

※ 패널티

1. 우주선 위치가 맵을 벗어난다.
2. 우주선 다리 위치의 맵 높이 차이가 6을 넘어간다.
3. 우주선이 겹치는 부분이 있다.

문제 분석

- 우주선 착륙시 점수에 영향을 미치는 상황

- 1) 착륙 가능한 땅이 겹쳐 있는 경우
- 2) 착륙 가능한 땅에 비해 우주선 재고가 부족한 경우

- 맵에 우주선을 착륙시킬 수 있는 땅은 정해져 있다.

착륙 가능한 땅은 몇 개쯤 될까?

- 착륙 가능한 땅과 우주선의 가로(h), 세로(w) 크기에 대해 $a=\min(h,w)$, $b=\max(h,w)$ 라고 정의하자.
땅과 우주선의 a, b가 같으면 착륙 가능하다.

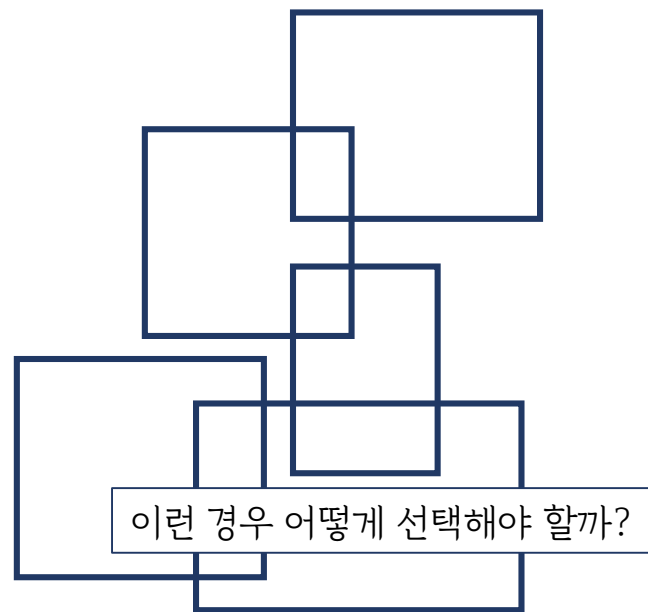
가능한 (a,b) 쌍의 개수는 (2,2) (2,3) (2,4) (2,5) (3,3) (3,4) (3,5) (4,4) (4,5) (5,5) 로 10개 뿐이다.

(a,b)가 같은 땅과 우주선은 각각 몇 개쯤 될까?

- 겹쳐짐을 고려하지 않으면 각각의 (a,b)에 대해 min(땅의 수, 우주선의 수) 개가 매칭된다.
겹쳐졌을 때의 처리가 관건으로 보인다.

착륙 가능한 땅의 분포를 파악해보자.

ex) 겹치지 않는 땅의 개수, 겹치는 땅이 최대(평균) 몇 개로 묶이는지 등



데이터 분석 결과

seed 1~50, tc 10개씩 확인

- 착륙 가능한 땅은 몇 개쯤 될까?

9000개 이하

- (a,b)가 같은 땅과 우주선은 각각 몇 개쯤 될까?

거의 모든 케이스에서 땅의 수 < 우주선의 수

(50개 seed * 10개 testcase) 중 6개에서만 땅의 수가 조금 더 많은 경우 발생

- 착륙 가능한 땅의 분포

겹쳐진 그룹 최대 크기 = 11(seed=5), 16(seed=1~50)

겹쳐진 그룹 최대 크기 평균 = 8(seed=1~50)

겹쳐지지 않는 땅의 수 = 5600 ~ 6000

※ 주목할 사실

- 우주선 재고가 부족해서 배정하기 힘든 경우는 극히 적다.
- 다른 땅과 겹쳐지지 않는 땅의 수가 절반을 넘어간다.
- 그룹에 속하는 땅의 개수가 그리 많지 않다.

ver1. 점수가 가장 큰 땅부터 배정한다.

seed=5, SCORE: 2994878
seed=32, SCORE: 3004560

가장 쉽게 떠올릴 수 있는 방법이다.

1. 우주선을 (a,b) 별로 분류한다.
2. 땅을 점수 순으로 정렬하여 가장 큰 땅부터 처리한다.

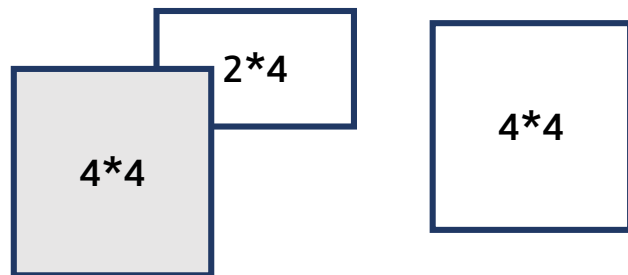
ver2. 그룹 크기가 1인 땅들 먼저 사용 + ver1

seed=5, SCORE: 2994878
seed=32, SCORE: 3004866

다른 땅이랑 겹치지 않는 땅들은 무조건 선택하는게 좋다.

그런 땅들을 먼저 처리하고 나면 우주선이 부족해서 손해보는 경우를 줄일 수 있다.

but, 대부분은 우주선 재고가 넉넉하여 극히 일부 케이스에서만 조금 개선된다.

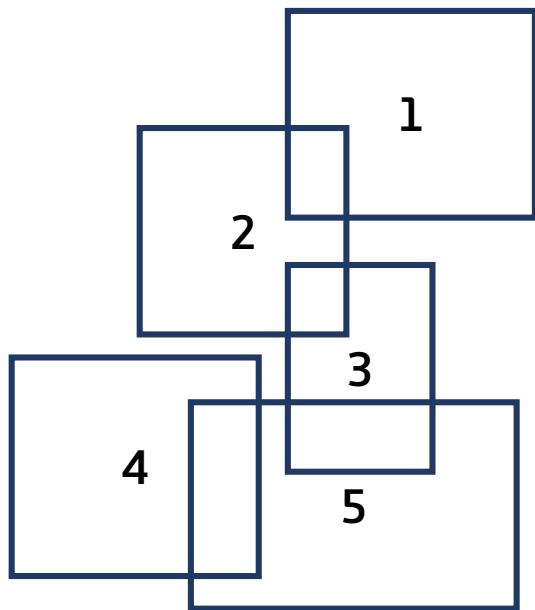


4*4, 2*4 우주선이 한 개씩 있을 때,
왼쪽 4*4 땅을 먼저 선택하는 경우를 방지할 수 있다.

ver3. 그룹별로 선택 가능한 모든 조합을 확인한다.

seed=5, SCORE: 3001673
seed=32, SCORE: 3011261

- 그룹에 속한 땅의 수(k)가 그리 많지 않다는 점을 활용한다.
- 모든 조합은 2^k 개 존재한다.
- 재귀를 이용하여 겹치지 않게끔 조합을 구성하고 그 중 최고 점수를 선택한다.



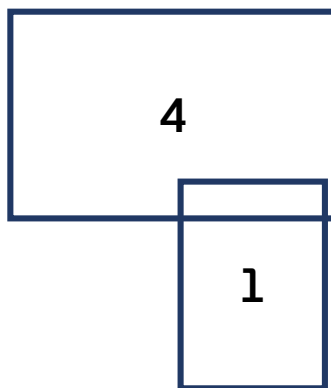
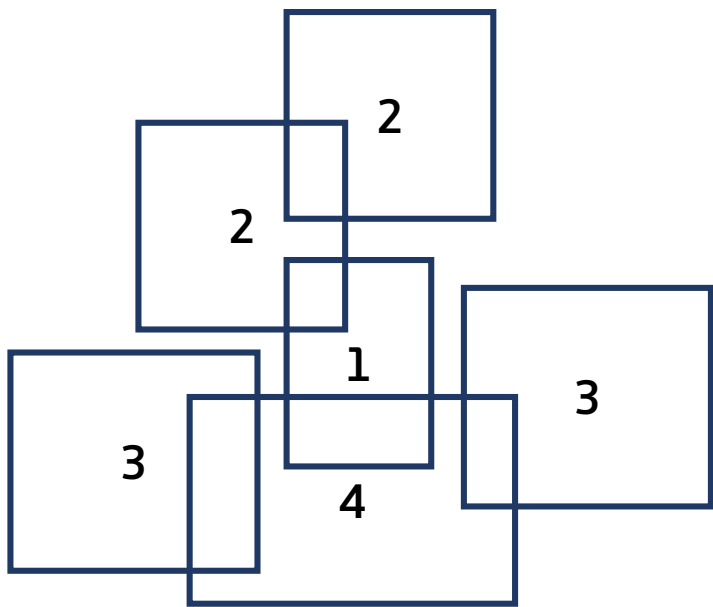
선택 가능한 조합

- 0개 : ()
- 1개 : (1) (2) (3) (4) (5)
- 2개 : (1,3) (1,4) (1,5) (2,4) (2,5) (3,4)
- 3개 : (1,3,4)
- 4개 : X
- 5개 : X

ver4. ver3+크기가 작은 그룹부터 처리한다.

seed=5, SCORE: 3001673
seed=32, SCORE: 3011679

- 크기가 1인 그룹은 무조건 먼저 처리하는게 좋다.
- 그룹 크기가 클수록 더 많은 조합이 가능하다.
- 특정 크기의 우주선 재고가 부족할 때, 그룹 크기가 클수록 더 나은 차선택이 있을 확률이 높지 않을까?



4가 없는 경우

- 왼쪽 그룹은 1, 2, 3, 3 을 선택하면 된다.
- 오른쪽 그룹은 1을 선택해야 한다.

구현 예시

두 사각형 겹치는지 판별

- 행, 열 각각 겹쳐야 됨
- $s1 \leq e2 \ \&\& \ s2 \leq e1$



그룹 생성

- union-find

그룹 크기순 정렬

- counting sort
- merge sort

조합

- 재귀 활용, 그룹 내 모든 땅에 대해 선택or선택X 두가지 경우를 전부 확인
- 선택 하는 경우는 재고가 남아있는지, 선택한 다른 땅과 겹치는지 판별 필요
- `recur(int idx, int score, int stat[])`
`recur(int idx, int score, int stat)`
- stat은 선택 여부 기록
배열로 하는 경우 $stat[idx] = 1/0$
정수로 하는 경우 $(stat \gg idx) \ \& \ 1 = 1/0$

땅 정보 저장

`land[i] = { sr, sc, er, ec, h, w, a, b, score }`

- (sr, sc) : 좌상단 좌표
- (er, ec) : 우하단 좌표
- h : 높이
- w : 너비
- a : $\min(h, w)$
- b : $\max(h, w)$
- score : $a * a * b$

(a,b)별로 우주선 분류

(a,b)	개수	우주선 번호
(2,2)	3	0, 3, 4
(2,3)	2	1, 2
..		
(5,5)	1	5

그룹별로 땅 분류

gid	개수	땅 번호
1		
2	2	2, 1
..		
n	1	n

gid=그룹 번호, n=땅의 수

그룹 크기순 정렬

idx	0	1	2	3
gid	n	4	2	5

그룹 크기 : $n \leq 4 \leq 2 \leq 5$

감사합니다