

아이디어

- 1. 퍼즐의 왼쪽 위에서부터 오른쪽 아래까지 퍼즐 조각들을 차례로 채워 넣는다고 할 때, N 개의 퍼즐 조각들을 채우는 방법은 $N!$ 가지의 경우가 있다.
- 2. 하나의 방법을 실행하는 중에 이미 채운 조각들의 집합을 `prefix`, 다음에 넣을 조각을 `p` 라 할 때, 다음 조각이 퍼즐에 들어갈 수 없다면 `prefix ∪ {p}` 를 `prefix`로 가지는 방법은 무시한다. (예를 들어 1번을 넣은 후 2번이 들어갈 수 없다면, 1, 2 로 시작하는 경우들은 해볼 필요가 없다.)

구현

순열을 만드는 recursive function인 `fit(x, y, remain[])` 을 이용한다. (x, y)는 다음 퍼즐 조각이 들어갈 위치를 나타낸다.

`remain`은 남은 퍼즐 조각의 집합이고, `remain`에 없는 퍼즐 조각들은 퍼즐을 나타내는 이차원 배열에 채워져 있다.

퍼즐의 첫 좌표인 (0, 0) 에서 시작하고, `remain`이 전체 퍼즐 조각의 집합인 `fit` 함수로 시작한다.

순열을 만들기 위해 `fit` 함수는 `remain`에 있는 퍼즐 조각을 골라 (x, y)에 넣을 수 있는지 확인한다.

선택한 조각을 퍼즐에 넣을 수 없다면 다른 조각을 찾고, 더 이상 남은 조각이 없는 경우 0을 리턴한다.

선택한 조각을 퍼즐에 넣을 수 있다면 넣고, 새로운 위치에 나머지 조각들을 넣는 `fit` 함수를 호출한다.

호출한 `fit` 함수가 0을 리턴하면 선택한 퍼즐 조각으로 퍼즐을 완성할 수 없는 것을 의미하므로, 퍼즐에서 뺀다.

남아 있는 마지막 조각이 퍼즐에 들어갈 때 퍼즐은 완성이 되고 `fit` 함수는 1을 리턴한다. 이 함수를 호출한 `fit` 함수도 1을 리턴하여 종료한다.

새로운 위치를 구하는 방법과 퍼즐이 들어갈 수 있는지 판별하는 방법

새 위치는 퍼즐의 y좌표가 빠르고 y좌표가 같다면 x좌표가 빠른, 아직 채워지지 않은 첫 번째 좌표(fig 1)로 정한다.

새 위치의 왼쪽과 위쪽 공간들은 채울 수 없기 때문에, 그 위치를 퍼즐 조각의 첫 번째 블록(fig 2)으로만 채울 수 있어 판별이 편리하다.

시간 복잡도

`fit` 함수는 최악의 경우 $N!$ 에 근접하게 호출 될 수 있다. `fit` 함수는 `remain`에 있는 모든 퍼즐 조각이 어떤 위치에 들어갈 수 있는지 확인한다.

따라서 각 퍼즐 조각 하나 당 어떤 위치에 들어갈 수 있는지 $N!$ 만큼 확인하므로 모든 퍼즐의 크기를 더한 값을 m 이라 하면 $O(m \times N!)$ 이 된다.

Fig 1.

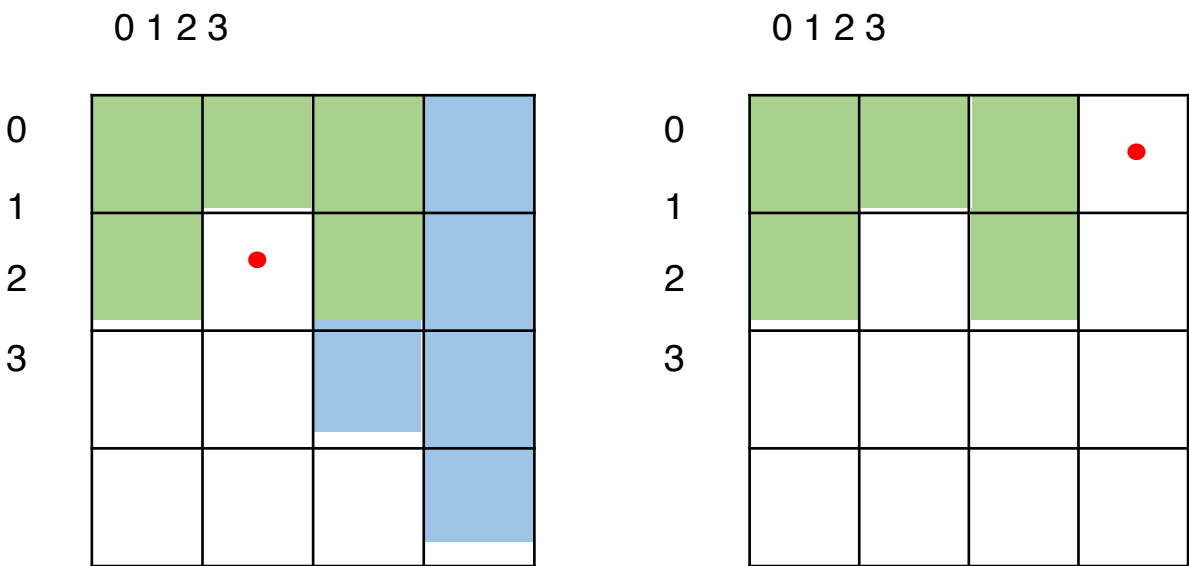


Fig 2.

