

1

트로미노(더 정확히 말하면 오른쪽 트로미노)는 3개의 1×1 제곱으로 형성된 L자 모양의 타일이다.

문제는 어떤 2개라도 커버하는 것이다.

$n \times 2$

n 트로미노가 있는 사각형의 체스판.

트로미노는 임의로 방향을 잡을 수 있지만, 모든 사각형을 가려야 한다.

중복 없이 정확히 누락된 것을 제외하고 탐색한다.[골94]

이 문제에 대한 분할 및 재귀 알고리즘 설계

This problem can be solved using Divide and Conquer. Below is the recursive algorithm.

// n is size of given square, p is location of missing cell

Tile(int n, Point p)

1) Base case: $n = 2$, A 2×2 square with one cell missing is nothing but a tile and can be filled with a single tile.

2) Place a L shaped tile at the center such that it does not cover the $n/2 \times n/2$ subsquare that has a missing square. **Now all four subsquares of size $n/2 \times n/2$ have a missing cell** (a cell that doesn't need to be filled). See figure 2 below.

3) Solve the problem recursively for following four. Let p_1, p_2, p_3 and p_4 be positions of the 4 missing cells in 4 squares.

a) Tile($n/2, p_1$)

b) Tile($n/2, p_2$)

c) Tile($n/2, p_3$)

d) Tile($n/2, p_4$)

The working of Divide and Conquer algorithm can be proved using Mathematical Induction. Let the input square be of size $2^k \times 2^k$ where $k \geq 1$.

Base Case: We know that the problem can be solved for $k = 1$. We have a 2×2 square with one cell missing.

Induction Hypothesis: Let the problem can be solved for $k-1$.

Now we need to prove to prove that the problem can be solved for k if it can be solved for $k-1$.

For k , we put a L shaped tile in middle and we have four subsquares with dimension $2^{k-1} \times 2^{k-1}$ as shown in figure 2 above. So if we can solve 4 subsquares, we can solve the complete square.

Time Complexity:

Recurrence relation for above recursive algorithm can be written as below. C is a constant.

$$T(n) = 4T(n/2) + C$$

The above recursion can be solved using Master Method and time complexity is $O(n^2)$

2

두 문자 문자열에는 공통적인 하위 문자열이 많이 있을 수 있다. 예를 들어, 사진 및 단층 촬영에는 길이 1(즉, 단일 문자) 및 공통의 몇 가지 하위 문자열이 있다.

서브스트링 **ph, to, o** 그래프(**Orgraph**의 모든 서브스트링) 최대 공통점 하위 문자열 길이는 6이다.

$X = x_1x_2 \dots x_n$ 과 $Y = y_1y_2 \dots y_m$ 은 두 문자열이 되게 한다. 동적 프로그래밍 제공 X 및 Y 에 대한 최대 공통 하위 문자열 길이를 찾기 위한 알고리즘. 의 순서는 무엇인가. 알고리즘 실행 시간?

3

부분집합의 합(**Subset Sum**) 문제는 다음과 같이 정의된다. ● 부분집합의 합(**Subset Sum**) 문제: 집합 S 는 자연수를 원소로 하며, 원소들의 개수가 유 한개이다.(원소가 무한히 많은 것이 아니라, 개수가 정해져 있다는 의미.) 집합 S 에 대해서, S 의 부분 집합들 중 원소의 합이 K 가 되는 부분집합을 찾으시오. (K 는 자연수) 이 부분집합의 합(**Subset Sum**) 문제가 **NP problem**임을 증명하시오