

1780_종이의 개수

정동욱

문제 설명

- 같은 번호로 이루어진 종이기가 아니면 9등분을 해서 같은 번호로 이루어진 종이기가 나올때까지 잘라서 각각의 번호만 쓰여진 종이의 갯수를 구하는 문제이다.

문제 해결

- 분할 정복 문제로 해당 조건을 만족하지 않으면 문제를 나눠서 해결하면 된다.
이 문제의 경우 한 번으로 이루졌는지 확인 후 아닐시 나누고 각각을 다시 확인하면 해결 가능하므로 재귀를 이용해서 답을 구할 수 있다.

핵심 코드

```
41 void func(int x,int y,int k){
42     // 한 번호로 이루어진 종이인지 확인
43     if(check(x,y,k)){
44         // 같은 번호로 이루어진 종이이면 해당 번호의 갯수를 증가
45         cnt[data[x][y]+1]++;
46         return;
47     }
48     int m=k/3;
49     for (int i=0; i<3; i++) {
50         for (int j=0; j<3; j++) {
51             // 재귀를 이용해서 한 번호가 될때까지 9등분한다.
52             func(x+i*m, y+j*m, m);
53         }
54     }
55 }
56
57 }
58
59 // 한 번호로 이루어진 종이인지 확인하는 함수
60 bool check(int x,int y,int k){
61     for(int i=x; i<x+k; i++){
62         for(int j=y; j<y+k; j++){
63             if(data[x][y] != data[i][j]){
64                 return false;
65             }
66         }
67     }
68     return true;
69 }
70
```

시간 복잡도

- 최악의 경우 $N \times N$ 크기의 종이를 크기가 1이 될때 까지 계속 9등분 해야하므로 한단계 분할 후 종이의 크기는 $N \times N / 9$ 가 된다.
즉, 몇단계까지 분할이 가능한지 살펴 보면 $\log_3 N$ 단계까지 발생한다.
여기서 각 단계에서 드는 연산량($O(1)$)의 총합은 $O(N)$ 이다.
- 그러므로, 시간복잡도는 $O(N \times \log_3 N)$ 이다.