

# week 9

2024-08-29

# labyrinth

The labyrinth is designed in such a way that there is exactly one path between any two free blocks

모든 두 점은 각각을 잇는 경로가 정확히 한 개 존재함→ 트리

트리의 지름 구하기

→ 탐색 두번으로 풀기

아무 점에서 탐색을 한번 해서 가장 먼 점을 구하고,  
구한 점에서 한번 더 탐색을 해 가장 먼 점을 구하면  
두번째 구한 점이 트리의 지름이 된다

# 색종이 붙이기

```
ans = go(now);
```

```
int x = now / 10;  
int y = now % 10;
```

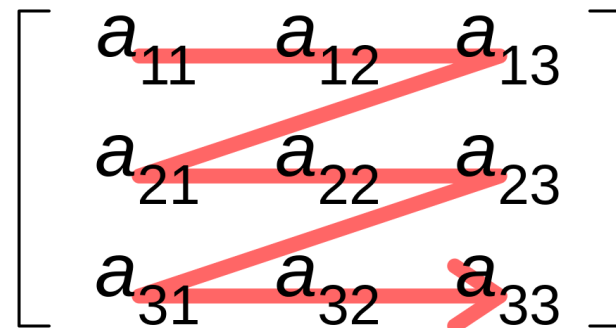
now 를 증가시키면 오른쪽으로 한 칸 가고  
행을 넘어가면 아래 열로 가서 첫 칸부터 시작

백트래킹 문제를 풀 때 복잡하지 않게 다음 칸으로 갈 수 있음

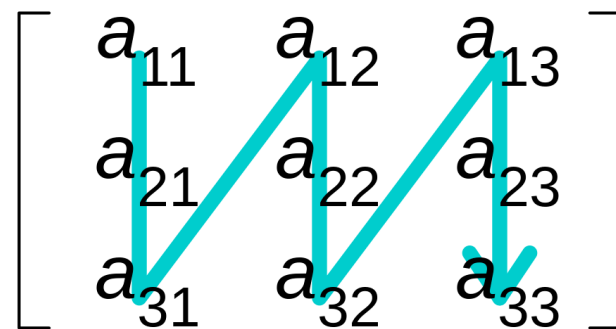
go(x, y) 라면 y 를 증가시키면서  $y == n$  일때  
go(x + 1, 0) 으로 가는 처리를 해야하고 불편함

탈출지점은  $now == 100$  일때임

Row-major order



Column-major order



# 색종이 붙이기

색종이 사이즈를 0~4 라고 했음  
그래서

$a[x][y] == 1$  이라면  
for 문을 돌며 0 에서 4 까지의 색종이를  
붙일 수 있는지 확인함  
( 색종이를 붙이면 10 이 넘어가는지 )

붙일 수 있는지 확인하면  
 $cnt[i]$  로 색종이가 남아있는 지 확인하고

fil 함수에서 색종이의 칸에 0 이 있는지 확인함  
붙일 칸이 전부 1 이라면 붙이고

색종이 갯수를 줄여줌  
그리고 정답을 갱신하고 돌아와서 색종이를  
다시 떼줌

```
int go(int now) {
    if(now == 100) {
        return 0;
    }
    int x = now / 10;
    int y = now % 10;
    int ret = 1e9;
    if(a[x][y] == 1) {
        for(int i = 0; i < 5; i++) {
            if(x + i >= 10 || y + i >= 10) break;
            if(cnt[i] > 0 && fil(x, y, i, -1)) {
                cnt[i]--;
                ret = min(ret, go(now + 1) + 1);
                fil(x, y, i, 1);
                cnt[i]++;
            }
        }
    } else {
        ret = go(now + 1);
    }
    return ret;
}
```

# 색종이 붙이기

fil 함수는 v 를 두어  
붙일때와 뗄 때 함께 사용할 수 있게 했음

V == -1 는 색종이를 붙이는 경우인데  
색종이 붙일 칸을 확인해서 0 이 있으면 false  
뗄 때는 경우에는 상관이 없음

그래서 v 값을 구간에 더해주면 됨

```
int a[10][10],
int cnt[5];
bool fil(int x, int y, int sz, int v) {
    if(v == -1) {
        for(int i = x; i <= x + sz; i++) {
            for(int j = y; j <= y + sz; j++) {
                if(a[i][j] == 0) return false;
            }
        }
    }
    for(int i = x; i <= x + sz; i++) {
        for(int j = y; j <= y + sz; j++) {
            a[i][j] += v;
        }
    }
    return true;
}
int go(int now) {
```

# 색종이 붙이기

진행하다가 불가능한 경우

$a[x][y] == 1$  인데 색종이가 없거나  
조건을 전부 다 만족하지 못한다면

$if(a[x][y] == 1)$  문을 그냥 나오게 되고

$1e9$  가 반환됨

그럼 거기서부터 이전으로 돌아가고  
붙인 색종이를 떼고 더 큰걸로 시도해봄  
큰것도 안되면 계속 뒤로 가면서

다른 방법을 시도하게됨

```
int go(int now) {
    if(now == 100) {
        return 0;
    }
    int x = now / 10;
    int y = now % 10;
    int ret = 1e9;
    if(a[x][y] == 1) {
        for(int i = 0; i < 5; i++) {
            if(x + i >= 10 || y + i >= 10) break;
            if(cnt[i] > 0 && fil(x, y, i, -1)) {
                cnt[i]--;
                ret = min(ret, go(now + 1) + 1);
                fil(x, y, i, 1);
                cnt[i]++;
            }
        }
    } else {
        ret = go(now + 1);
    }
    return ret;
}
```