

3114: 사과와 바나나

컴퓨터학부 20142314 김승호

발표 순서

- 문제 요약
- 해결 방법 찾기
- 코드로 옮기기
- Q&A

문제 요약

- A나라 vs B나라
- A나라가 좋아하는 것은 사과
- B나라가 좋아하는 것은 바나나
- 불도저로 국경선을 정함
- A나라의 사과 + B나라의 바나나를 최대한으로

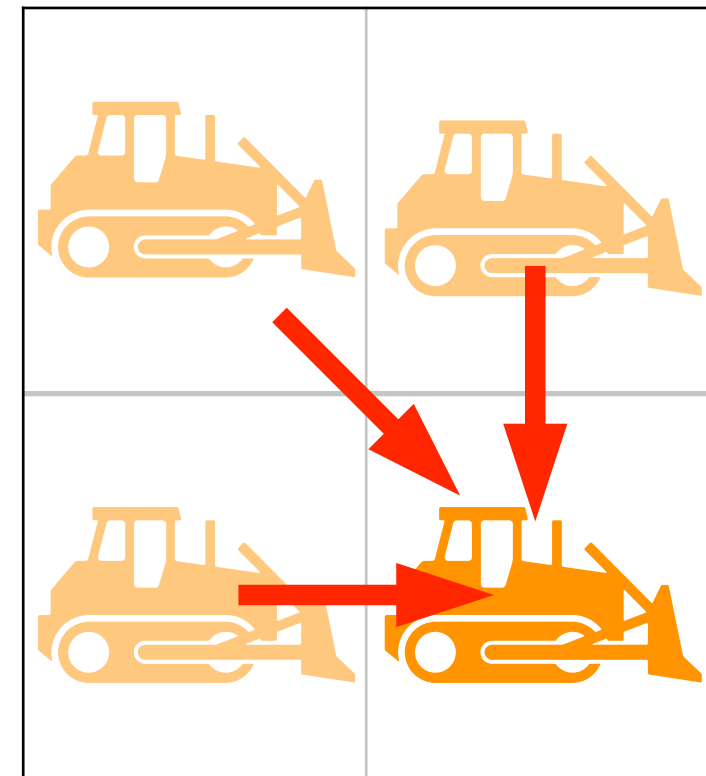
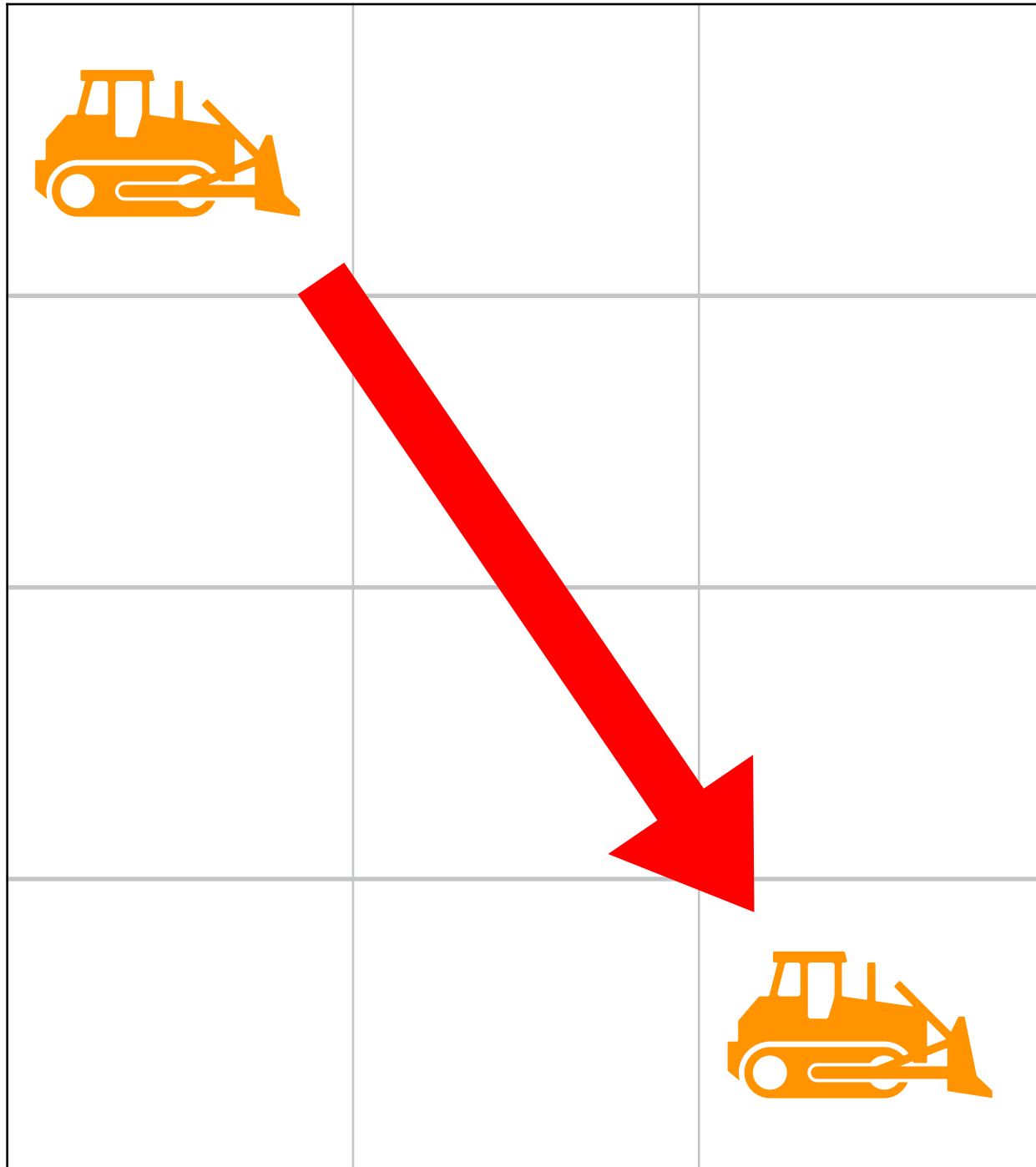


B나라



A나라

문제 요약 - 불도저가 움직이는 방법



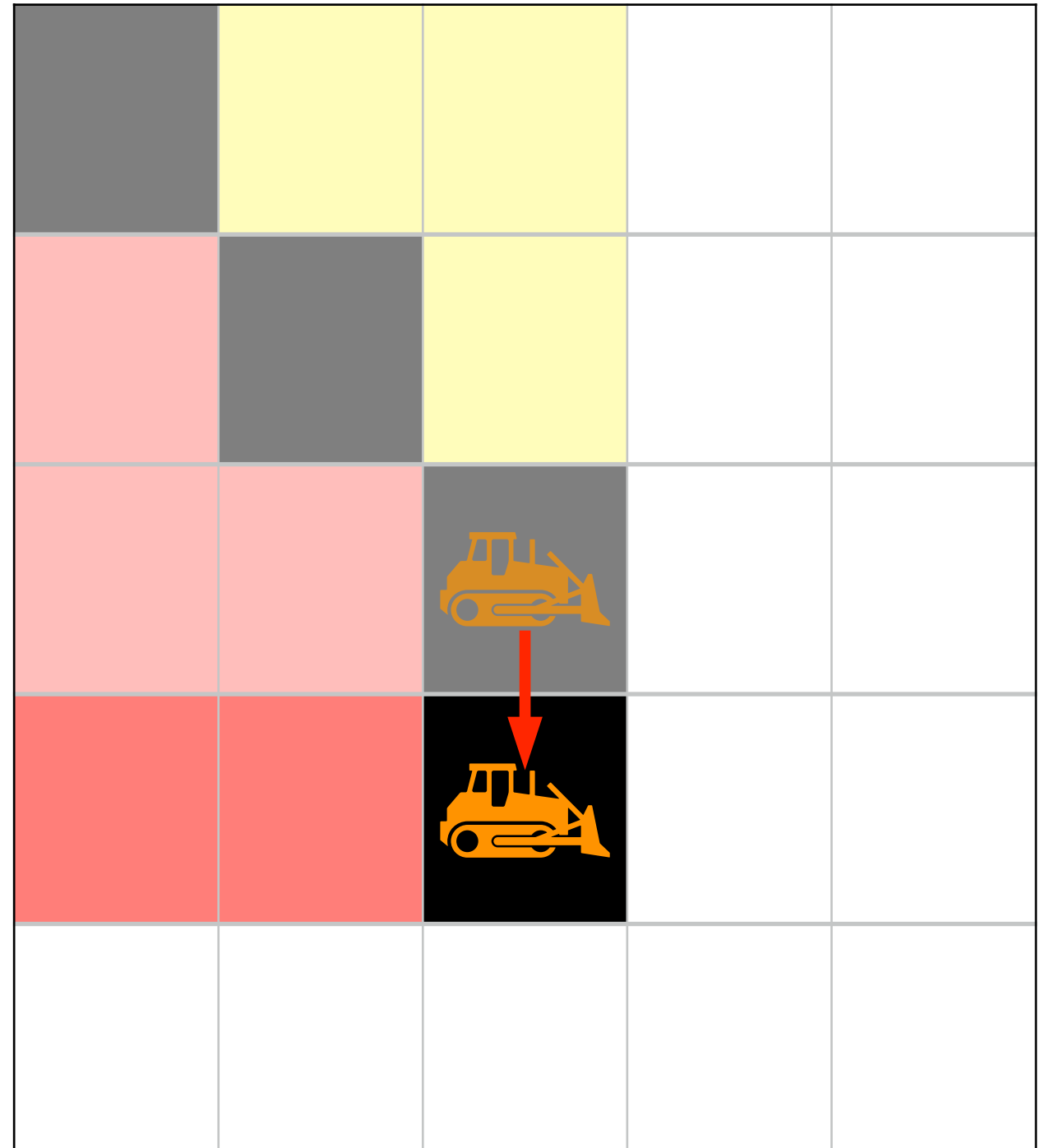
해결 방법 찾기

- 국경선은 계산에 포함 X
- 국경선 아래는 사과만
- 국경선 위는 바나나만

B2	B3	B5
A3	B1	A1
A2	A4	B1
B1	B3	A3

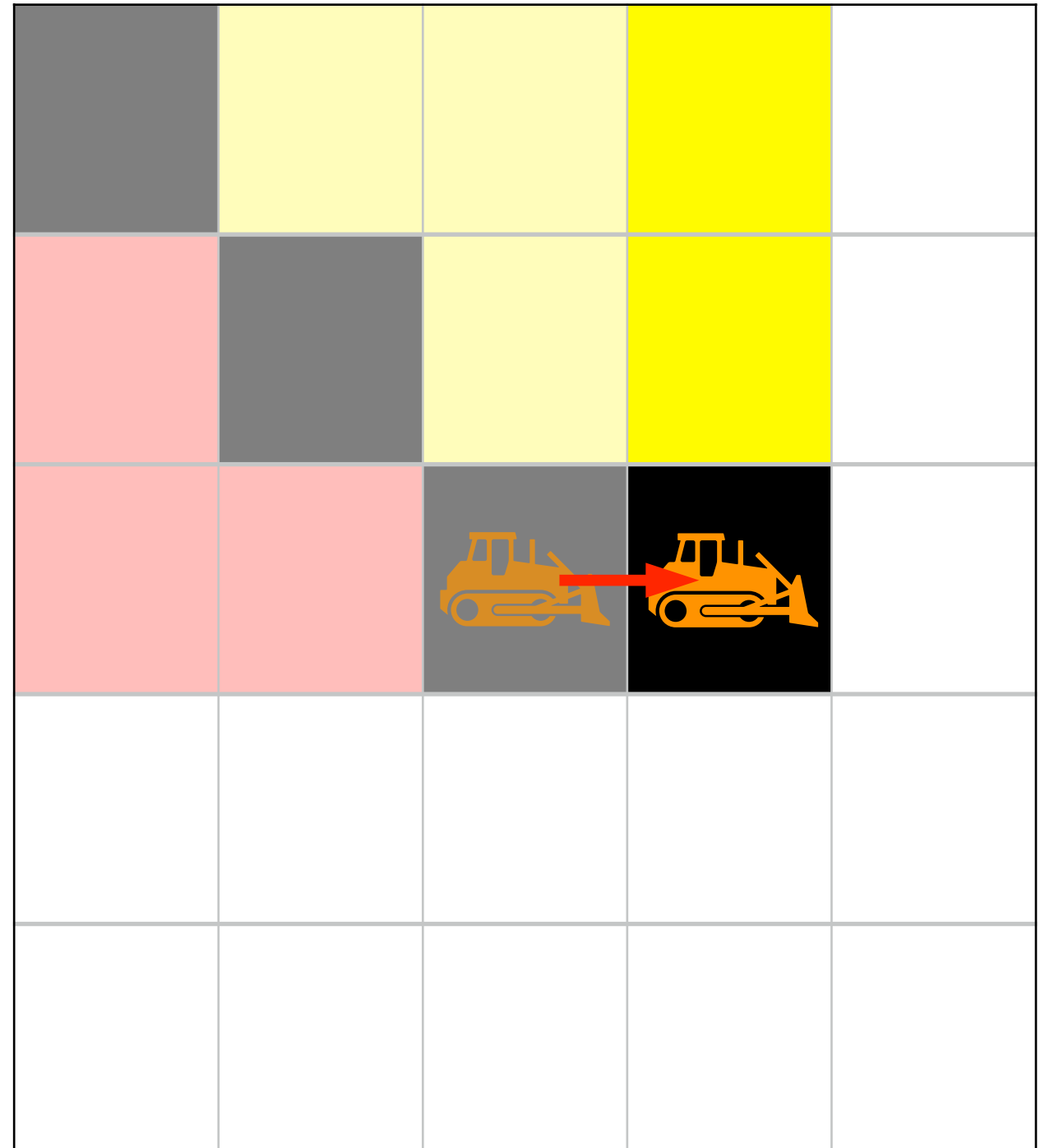
해결 방법 찾기 - 케이스 #1

- 불도저가 아래쪽으로 이동
 $(3, 3) \rightarrow (3, 4)$
- 4행 [1,2]열의 '사과' 개수 추가
- 일반화
 $(x, y) \rightarrow (x, y+1)$ 일 때,
 $(y + 1)$ 행 [1, x)열의
'사과' 개수 추가



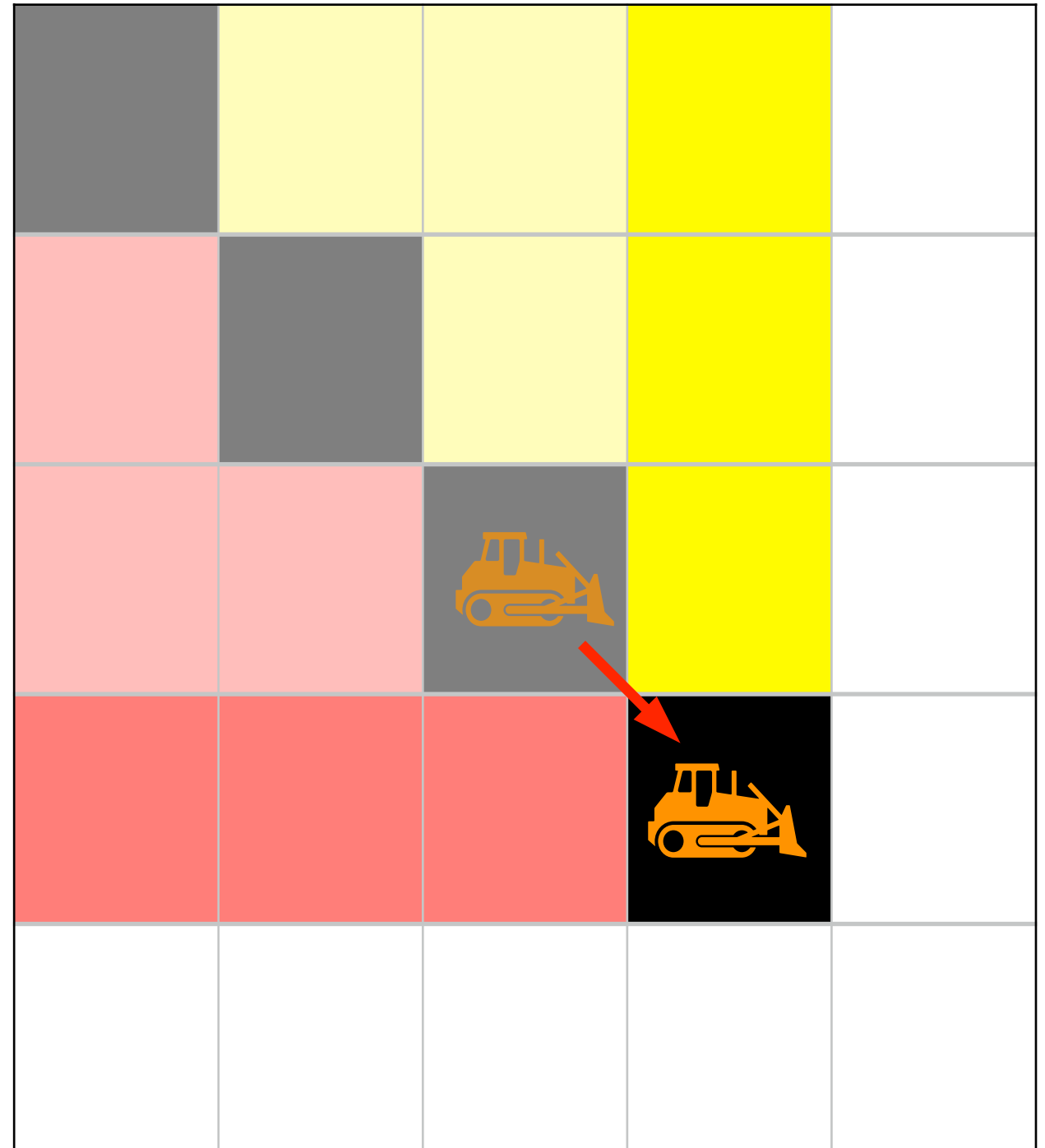
해결 방법 찾기 - 케이스 #2

- 불도저가 오른쪽으로 이동
 $(3, 3) \rightarrow (4, 3)$
- 4열 [1,2]행의 '바나나' 개수 추가
- 일반화
 $(x, y) \rightarrow (x+1, y)$ 일 때,
 $(x+1)$ 열 [1, y)행의
'바나나' 개수 추가



해결 방법 찾기 - 케이스 #3

- 불도저가 오른쪽 아래 대각선으로 이동
 $(3, 3) \rightarrow (4, 4)$
- 4열 [1,3]행의 '바나나'
4행 [1,3]열의 '사과'
개수 추가
- 일반화
 $(x, y) \rightarrow (x+1, y+1)$ 일 때,
 $(x + 1)$ 열 [1, $y+1$)행의 '바나나' 개수
 $(y + 1)$ 행 [1, $x+1$)열의 '사과' 개수
추가



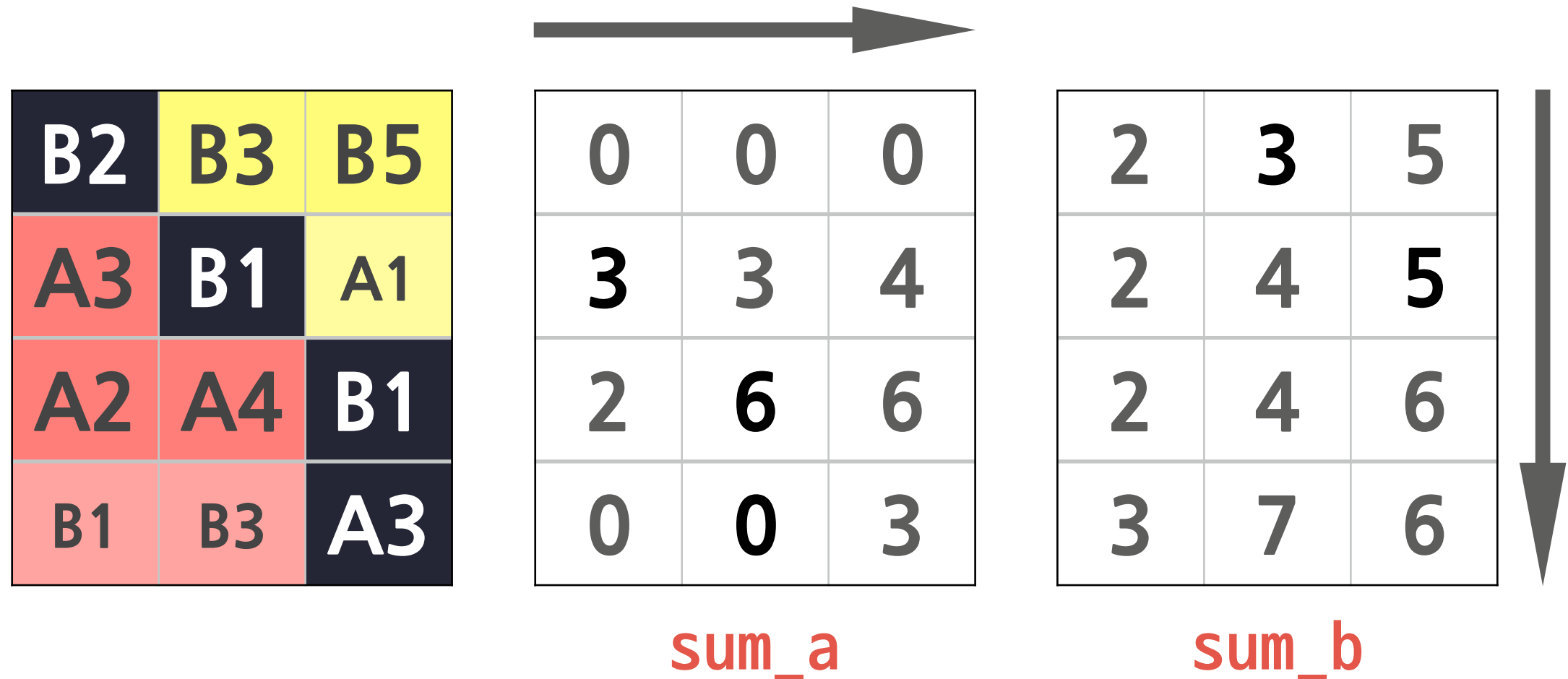
해결 방법 찾기 - 점화식 만들기

i열 j행에 대하여

```
solve(i, j) = max(  
    solve(i-1, j) + sum_a[i][j-1] (if i > 1),  
    solve(i, j-1) + sum_b[i-1][j] (if j > 1),  
    solve(i-1, j-1) + sum_a[i][j-1] + sum_b[i-1][j]  
                                                (if i > 1 && j > 1)  
)
```

```
solve(1, 1) = 0
```

해결 방법 찾기 - sum_a, sum_b 예제



$$3 + 6 + 0 + 3 + 5 = 17$$

코드로 옮기기

```
#include <algorithm>
```

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
struct tree {
```

```
    char type;
```

```
    int count;
```

```
};
```

```
tree arr[1501][1501];
```

```
int r, c, sum_a[1501][1501] = {}, sum_b[1501][1501] = {},
```

```
    dp[1501][1501];
```

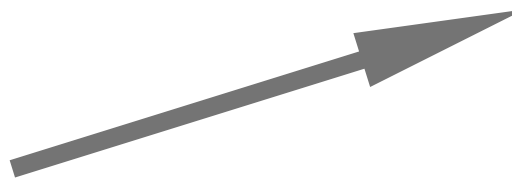
```
int main() {
    ios::sync_with_stdio(false);
    cin.tie(NULL);

    cin >> r >> c;
    fill_n(&dp[0][0], 1501 * 1501, -1);
```

```
for (int i = 1; i <= r; ++i) {
    for (int j = 1; j <= c; ++j) {
        string buf;
        cin >> buf;
```

```
arr[i][j].type = buf[0];
arr[i][j].count =
    stoi(buf.substr(1, buf.length() - 1));
```

type	count
A	3



```
arr[i][j].type = buf[0];
```

```
arr[i][j].count =
```

```
    stoi(buf.substr(1, buf.length() - 1));
```

```
sum_a[i][j] = sum_a[i][j - 1] +
```

```
    (buf[0] == 'A' ? arr[i][j].count : 0);
```

```
sum_b[i][j] = sum_b[i - 1][j] +
```

```
    (buf[0] == 'B' ? arr[i][j].count : 0);
```

```
}
```

```
}
```

```
cout << solve(r, c) << endl;
```

```
}
```

```
int solve(const int &i, const int &j) {  
    int &ret = dp[i][j];  
  
    if (ret != -1)  
        return ret;  
  
    if (i == 1 && j == 1)  
        return 0;  
  
    if (i > 1)  
        ret = max(ret, solve(i - 1, j) + sum_a[i][j - 1]);  
  
    if (j > 1)  
        ret = max(ret, solve(i, j - 1) + sum_b[i - 1][j]);  
  
    if (i > 1 && j > 1)  
        ret = max(ret, solve(i - 1, j - 1) +  
            sum_a[i][j - 1] + sum_b[i - 1][j]);  
  
    return ret;  
}
```

Q & A

감사합니다.