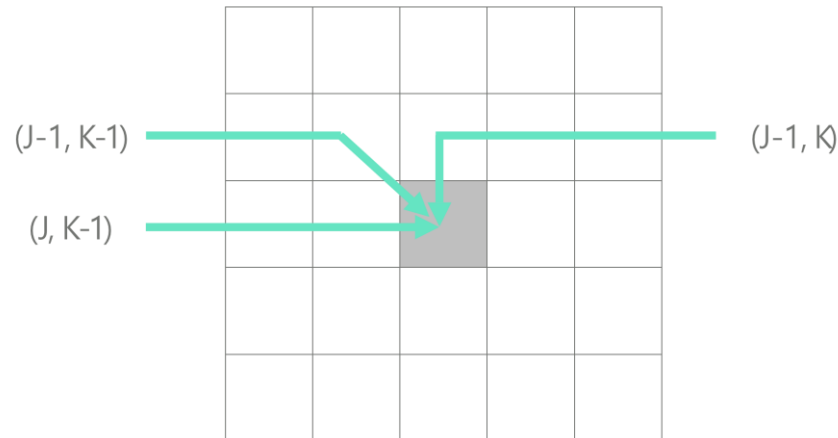
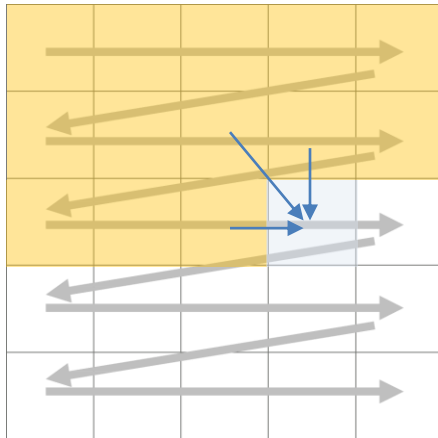


이동하기

<https://www.acmicpc.net/problem/11048>

- $dp[i][j] = (1, 1) \rightarrow (i, j)$ 로 이동하였을 때 가져갈 수 있는 사탕의 최댓값
 - ① $dp[i-1][j] + A[i][j]$
 - ② $dp[i][j-1] + A[i][j] \rightarrow \max(①, ②, ③)$
 - ③ $dp[i-1][j-1] + A[i][j]$
- 항상 아래와 오른쪽으로만 갈 수 있으므로 (i, j) 에서 가능한 이동은 $(i+1, j)$, $(i, j+1)$, $(i+1, j+1)$ 이 된다. 아래와 같이 dp 테이블을 for문을 돌면서 채워나가면 3 방향으로 dp값을 갱신할 수 있다.



이동하기

<https://www.acmicpc.net/problem/11048>

```
for (int i = 1; i <= N; ++i) {  
    for (int j = 1; j <= M; ++j) {  
        dp[i][j] = max({ dp[i - 1][j], dp[i][j - 1], dp[i - 1][j - 1] }) + A[i][j];  
    }  
}
```

- $i-1, j-1$ 범위 검사를 하지 않은 이유
 $i = 1, j = 1$ 인 경우 : $dp[i-1][j], dp[i][j-1], dp[i-1][j-1]$ 은 0이기 때문
 $i = 1$ 인 경우 : $dp[i-1][j] = 0 < dp[i][j-1]$ 이기 때문
 $j = 1$ 인 경우 : $dp[i][j-1] = 0 < dp[i-1][j]$ 이기 때문
- 이 문제의 경우 $A[i][j] \geq 0$ 이기 때문에 대각선의 요소는 고려할 필요가 없었다.

3	4
1	2

3+2 보다 (3+1+2) 또는 (3+4+2)의 값이 더 크다.

만약 음수 값이 들어올 수 있다면 대각선의 요소는 고려되어야 한다.

이동하기

<https://www.acmicpc.net/problem/11048>

```
vector<vector<int>> dp2, A2;
```

```
int solve(int y, int x) {  
    if (y == 1 && x == 1) return A2[1][1];  
    if (y < 1 || x < 1) return 0;  
  
    if (dp2[y][x] >= 0) return dp2[y][x];  
  
    dp2[y][x] = max({ solve(y - 1, x), solve(y, x - 1), solve(y - 1, x - 1) }) + A2[y][x];  
    return dp2[y][x];  
}
```

기저

참조 불가능한 값

메모이제이션

