

# F-グリッドの番号

原案: monkukai

問題文: monkukai

解説: monkukai

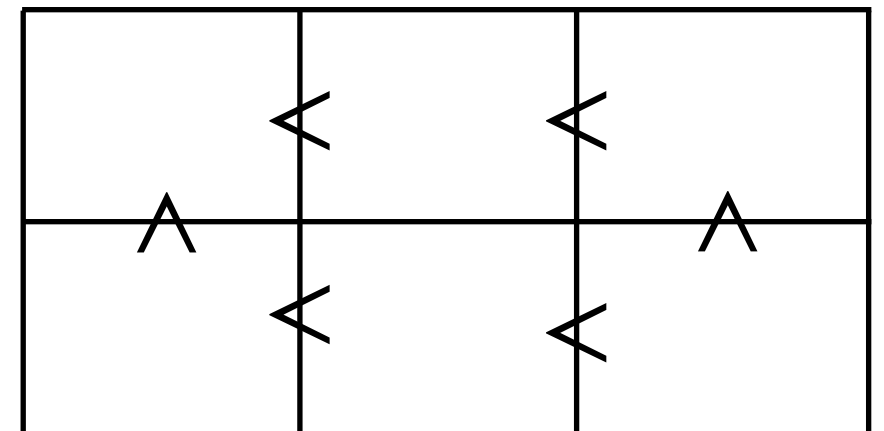
# 問題概要

- 1 から  $2 \times n$  までの整数をグリッドに書き込む.
- 以下の二つのルールを守る必要がある.
  - ① 図のような不等式を満たす
  - ② 隣り合う整数の差を  $k$  以下にする
- 整数の書き込み方は全部で何通りか?

[制約]

$$1 \leq n \leq 100$$

$$1 \leq k \leq 10$$



# 問題概要

- 1 から  $2 \times n$  までの整数をグリッドに書き込む.

- 以下の二つのルールを守る必要がある.

① 図のような不等式を満たす

② 隣り合う整数の差を  $k$  以下にする

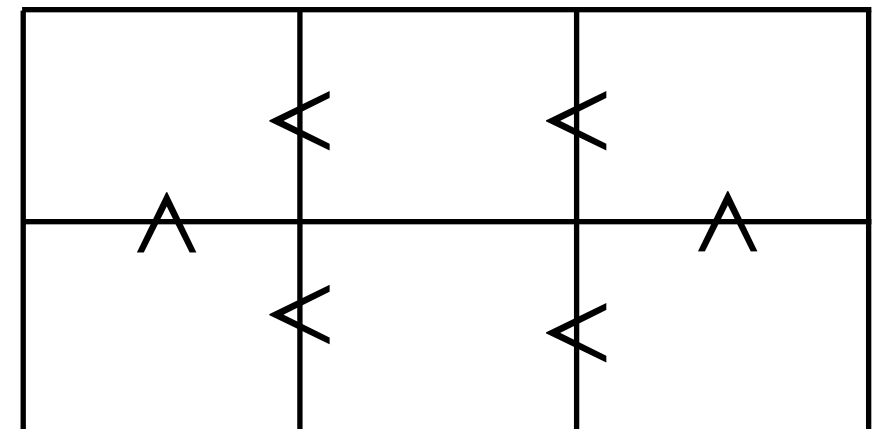
一旦、ルール①  
のみを考えてみる

- 整数の書き込み方は全部で何通りか?

[制約]

$$1 \leq n \leq 100$$

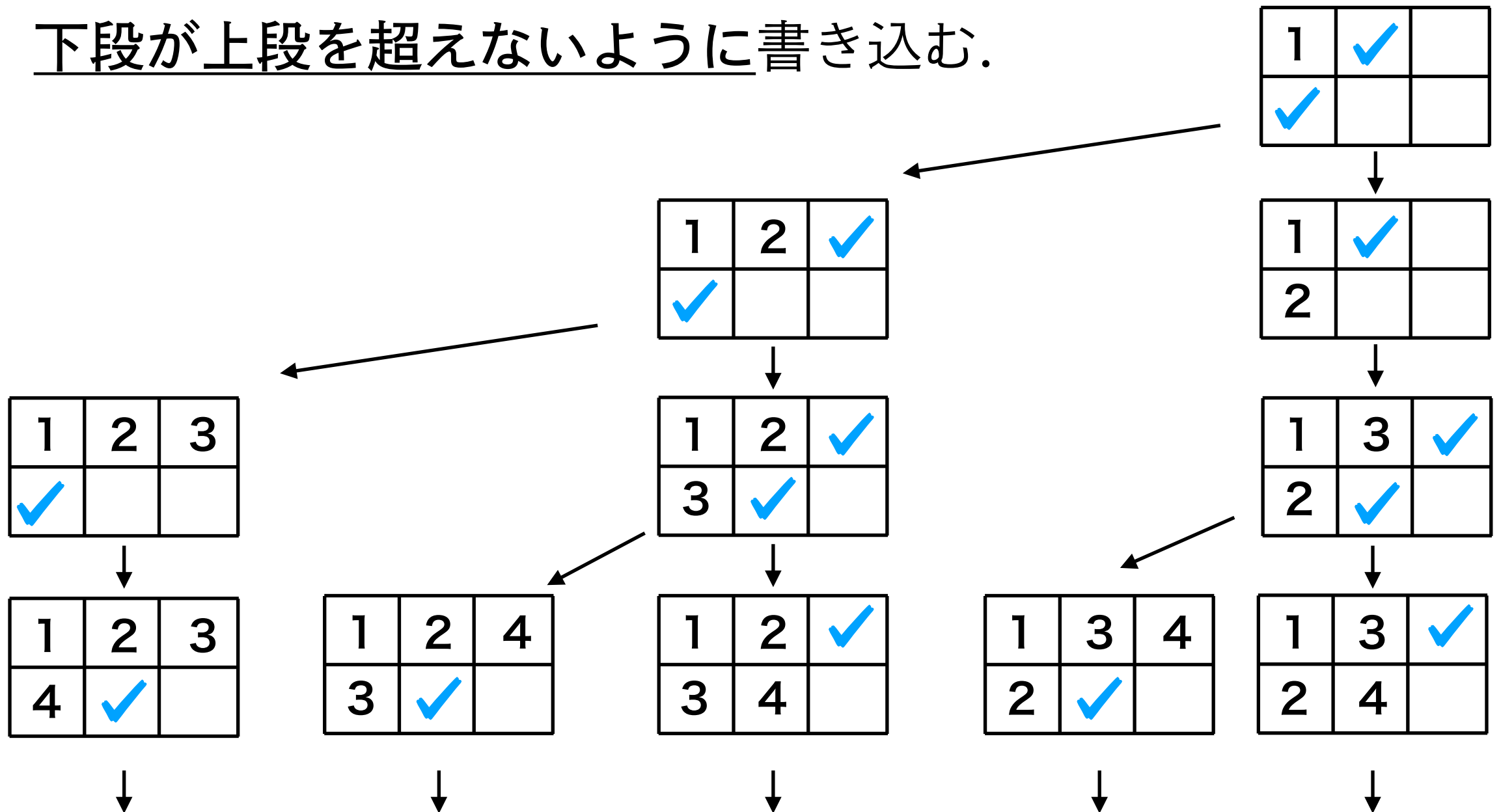
$$1 \leq k \leq 10$$



# $n = 3$ の例

✓ : 次ステップで書き込める場所

- 整数を 1 から順番に, 隙間ができないように,  
下段が上段を超えないように書き込む.



# 考察(1/2)

① 図のような不等式を満たす

② 隣り合う整数の差を  $k$  以下にする

- 条件①だけの問題なら、

$dp[i][j] :=$  整数  $i$  まで書き込み済みで、

上段が  $j$  個埋まっている時の通り数

で  $O(n^2)$  でこの問題が解ける。

1	<	2	<	3
^		^		^
4	<	✓	<	

1	<	3	<	4
^		^		^
2	<	✓	<	

# 考察(1/2)

- 条件①だけの問題なら、

$dp[i][j] :=$  整数  $i$  まで書き込み済みで、

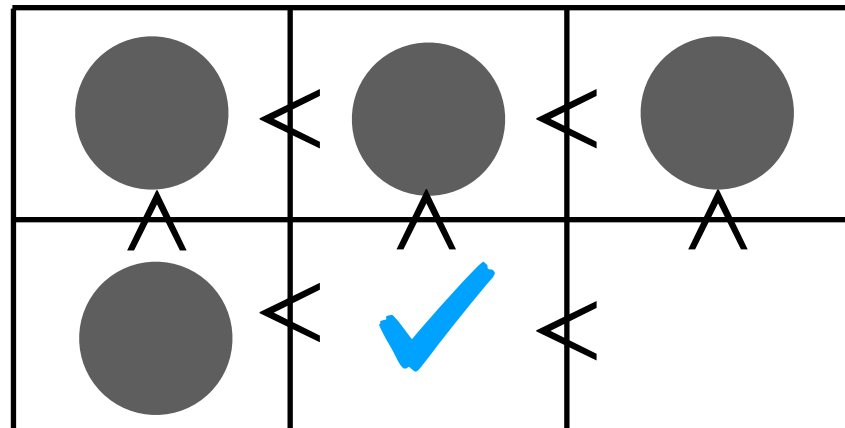
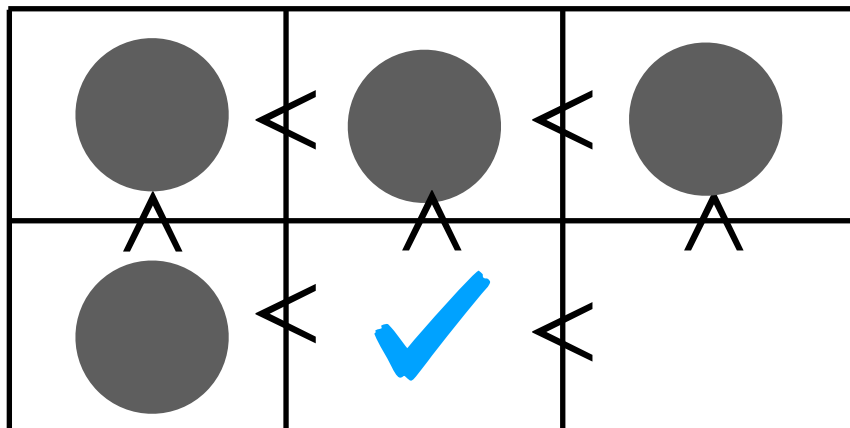
上段が  $j$  個埋まっている時の通り数

で  $O(n^2)$  でこの問題が解ける。

① 図のような不等式を満たす

② 隣り合う整数の差を  $k$  以下にする

状態をまとめることができる



$dp[4][3]$

に対応

# 考察(2/2)

- 条件②を考慮するには、  
今後、隣り合う可能性のあるマスを覚えておく必要がある。

- ① 図のような不等式を満たす
- ② 隣り合う整数の差を  $k$  以下にする

●	<	●	<	3	<	5	<	7	<	✓
^		^		^		^		^		^
●	<	6	<	✓	<		<		<	

$n = 6$  の例

# 考察(2/2)

- 条件②を考慮するには、  
今後、隣り合う可能性のあるマスを覚えておく必要がある。

- ① 図のような不等式を満たす
- ② **隣り合う整数の差を  $k$  以下にする**

赤色の波線部分の状態を覚えて  
DP をすれば良い

●	<	●	<	3	<	5	<	7	<	✓
Λ		Λ		Λ		Λ		Λ		Λ
●	<	6	<	✓	<		<		<	
Λ		Λ		Λ		Λ		Λ		Λ

$n = 6$  の例



# 想定解法

$dp[i][state] :=$  整数  $i$  まで書き込み済みで、  
赤波線部分が  $state$  な時の通り数.

右上に書き込む

●	●	3	5	7	✓
●	6	✓			

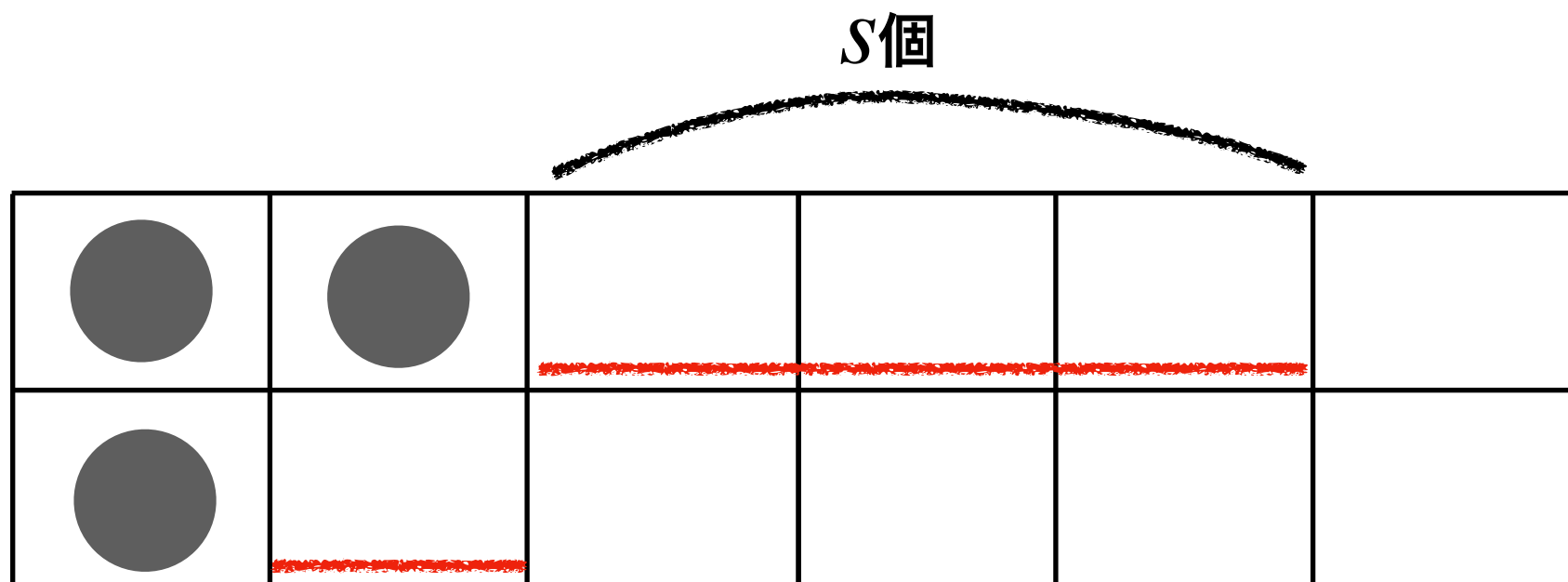
左下に書き込む

●	●	3	5	7	8
●	6	✓			

●	●	●	5	7	✓
●	●	8	✓		

# 計算量解析

- ・ 次に置く整数を  $i$  とする.
- ・ 上段と下段に置かれている整数の個数の差を  $s$  と置く.
- ・ 赤線部分に置かれる整数は  $i - k$  から  $i - 1$  までの  $k$  種類.
- ・ さらに上段の赤線部分には昇順に置くことを踏まえると, 状態数は  ${}_{k-1}C_s \times (s + 1)$  となる.



# 計算量解析

- よって  $i$  を固定した時の状態数の総和は

$$\sum_{s=0}^{k-1} {}^{k-1}C_s \times (s+1) < k \sum_{s=0}^{k-1} {}^{k-1}C_s = k \times 2^{k-1}$$

で押さえられる.

- DP の遷移は定数回であるので,  
アルゴリズム全体の時間計算量は  $O(nk2^k)$  となる.

# Writer 解

- **monkukui (C++) 90 行**
- **Tsuta\_J (C++) 106 行**
- **Tsuta\_J (python) 90 行**
- **TAB (C++) 45 行**
- **kazu (C++) 107 行**

# 提出状況

- **On-site: honunokoibito (120:23)**
- **On-line: ICPC\_MockaidoUniv (81:49)**
- **18 / 85 (21.17 %)**