

# Конструктивни задачи

Национална школа по информатика 26.08.2026г.

Изготвил: Велислав Гърков

# Задача за конструиране на пермутация

- По дадено  $N$  отпечатайте пермутация  $P$  на числата от 1 до  $N$  такава, че да НЕ съществува тройка  $(i, j, k)$ , където  $1 < i < j < k$  и  $P_j - P_i = P_k - P_j$ .
- Валидни пермутации:  $\{3, 5, 4, 1, 2, 6\}$ ,  $\{5, 1, 6, 3, 2, 4\}$ .
- Невалидни пермутации:  $\{1, \underline{4}, 3, 2, \underline{5}, \underline{6}\}$ ,  $\{\underline{6}, 3, \underline{4}, 1, \underline{2}, 5\}$ .

# Когато не знаем как да подходим

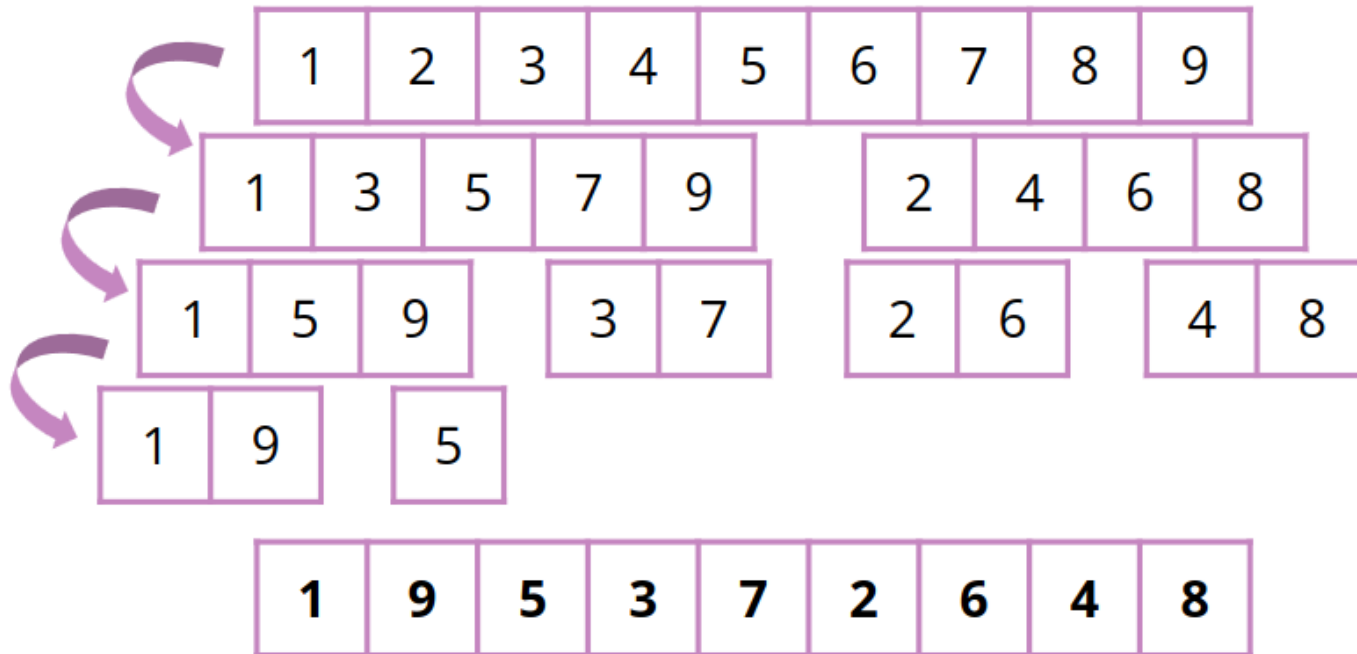
- Разписваме/намираме с brute force решения за малките случаи.
- Опитваме се да намерим някаква зависимост.
- Опитваме се да познаем конструкцията
- Пробваме да докажем/опровергаем предположението си

•В повечето решения числата са групирани по четност. Защо това помага за намирането на такава пермутация? Как да продължим след такава разделяне?

| 4       | 5         | 6           | 8               | 9                 |
|---------|-----------|-------------|-----------------|-------------------|
| 1 3 2 4 | 1 5 3 2 4 | 1 5 3 2 6 4 | 1 5 3 2 7 6 4 8 | 1 9 5 3 2 7 6 4 8 |
| 1 3 4 2 | 1 5 3 4 2 | 1 5 3 4 2 6 | 1 5 3 2 7 6 8 4 | 1 9 5 3 2 7 6 8 4 |
| 2 1 4 3 | 2 1 4 5 3 | 1 5 3 4 6 2 | 1 5 3 7 2 6 4 8 | 1 9 5 3 7 2 6 4 8 |
| 2 4 1 3 | 2 4 1 5 3 | 1 5 3 6 2 4 | 1 5 3 7 2 6 8 4 | 1 9 5 3 7 2 6 8 4 |
| 2 4 3 1 | 2 4 3 1 5 | 1 5 6 3 2 4 | 1 5 3 7 4 2 8 6 | 1 9 5 3 7 4 2 8 6 |
| 3 1 2 4 | 2 4 3 5 1 | 2 1 6 4 5 3 | 1 5 3 7 4 8 2 6 | 1 9 5 3 7 4 8 2 6 |
| 3 1 4 2 | 2 4 5 1 3 | 2 6 1 4 5 3 | 1 5 3 7 4 8 6 2 | 1 9 5 3 7 4 8 6 2 |
| 3 4 1 2 | 3 1 2 5 4 | 2 6 4 1 5 3 | 1 5 3 7 6 2 4 8 | 1 9 5 3 7 6 2 4 8 |
| 4 2 1 3 | 3 1 5 2 4 | 2 6 4 3 1 5 | 1 5 3 7 6 2 8 4 | 1 9 5 3 7 6 2 8 4 |
| 4 2 3 1 | 3 1 5 4 2 | 2 6 4 3 5 1 | 1 5 3 7 6 8 2 4 | 1 9 5 3 7 6 8 2 4 |
|         | ...       | ...         | ...             | ...               |

# Предположение

• Предполагаем, что след като разделим числата спрямо четността, можем да продължим да разделяме двете групи по остатък спрямо 4 и т.н.



# Доказателство

- Разделянето на числата по четност помага значително, защото така ако съществува тройка  $(i,j,k)$ , то  $|P_j - P_i| \neq 1$ . Тоест тройката индекси се намира или **изцяло** в лявата половина(да кажем при нечетните) или **изцяло** в дясната половина.
- Достатъчно е да разгледаме само една половина. Разликата между всеки две числа се дели на **2**. Затова след компресия на числата(получаваме пермутация на числата от 1 до  $N/2$ ), се свежда до

# Код на Г(р)ей

.Искаме да подредим числата  
от 0 до  $2^N-1$  включително в  
кръг така, че всеки 2 съседни  
числа да се различават с  
точно 1 бит.

10

11

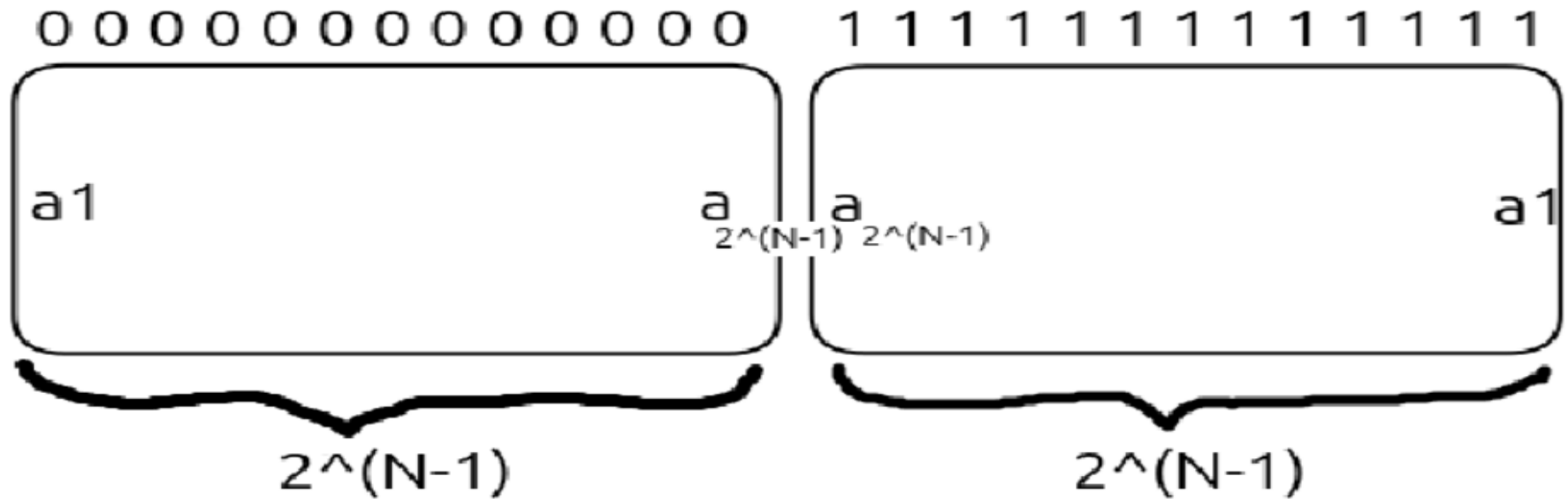
01

00

.0, 1

.00, 01, 11, 10

.000, 001, 011, 010, 110, 111, 101, 100



# IZHO 2019 The “Lyuboun” code

Иска се същото както при кода на Грей,  
но всеки 2 съседни числа да се различават с  
точно  $K$  бита. Ако е невъзможна  
такава конструкция, да се изведе  $-1$ .

•Помислете първо за кои стойности на  $K$  е  
невъзможно и защо.

# IZHO 2019 The “Lyuboyu” code

• Ако  $K$  е четно и започнем конструкцията с число  $x$ , броят на битовете на следващите числа винаги ще бъде със същата четност като броя на битовете в  $x$ . Следователно няма как да минем през всички числа от  $0$  до  $2^N - 1$ .

• Ако  $K \geq N$ , също е невъзможно.

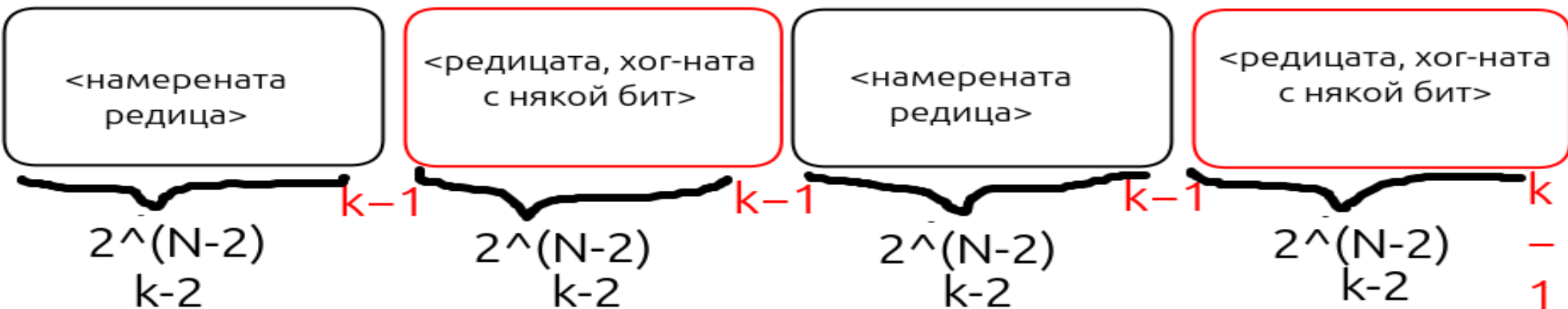
• Да се опитаме да решим рекурсивно задачата.  
Вече имаме конструкция за  $K = 1$ , как можем да я

# IZHO 2019 The “Lyuboyun” code

.Нека сме намерили редицата за  $N-2$ ,  $k-2$ .  
Отгоре върху редицата можем да редуваме 00, 11, както и 10, 01, като така се постига разлика от  $k$  бита. Как да свържем тези сегменти, където редуваме 00, 11 с тези, в които редуваме 10, 01?

# IZHO 2019 The “Lyuboyun” code

Хор-ването на редицата с даден бит не променя разликата в битовете и продължава да съдържа всички  $2^{(N-2)}$  числа. Затова хор-ваме редицата с бит който е различен в първото и последното число



# Една по-лесна задача

• <https://codeforces.com/contest/763/problem/B>

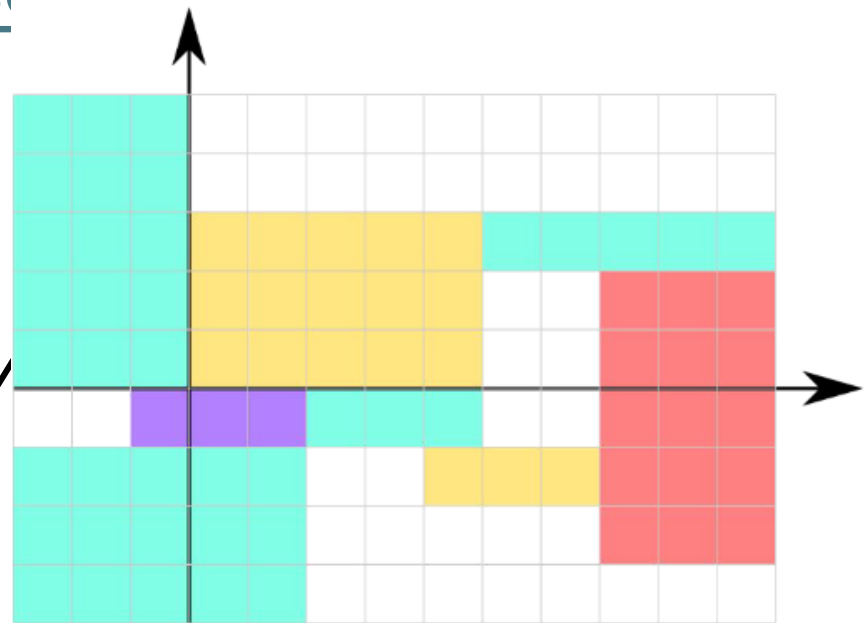
• Вход:

• - Непресиращи се правоъгълници

• - координати с цели числа

• - страни с **нечетна** дължина

• Изход:



# Наблюдение

• Да вземем долния ляв ъгъл на всеки правоъгълник. Какво можем да кажем за х-овете на тези ъгли на правоъгълници, които се допират? Какво следва от това?

# Наблюдение

- Ако два правоъгълника се допират, то  $x$ -овете на долните им леви ъгли са с различна четност.
- Същото можем да кажем и за  $y$ -ците.
- Можем да оцветим правоъгълниците на база на четността на  $x$ -а и  $y$ -а на долните им леви ъгли и така винаги ще имаме валидно оцветяване.

# Задача с дърво

• <https://codeforces.com/problemset/problem/1098/C>

• По дадено  $N$  и  $S$  да се конструира дърво с минимален разклонителен коефициент със сума на размерите на поддърветата  $S$ . Ако е невъзможно, да се изведе No.

• Помислете кои стойности на  $S$  са възможни и кои не са. Измислете механизъм да стигнете от една стойност до друга по лесен начин.

# Наблюдения

- Сумата от големините на поддърветата можем да я представим като сума на дълбочините на всеки връх (започвайки от дълбочина 1 при корена).
- Така лесно се вижда, че максималната сума е тази при дърво-пръчка.
- Минималната сума пък се постига при дърво звезда.
- Интересува ни не как изглежда дървото, а колко

# Непрекъснатост

• Ако искаме да намалим сумата, можем да хванем най-дълбокия връх и да го придвижим по-близко до корена (тоест да намалим дълбочината му). Качваме върха колкото е разликата между текущата сума и  $S$ . Ако можем да го направим дете на корена и сумата да остане по-голяма от исканата, го слагаме там и продължаваме със следващия най-дълбок връх.

• Това ни дава свойството непрекъснатост – можем да постигнем всяка възможна сума от минималната

# Решение

.Остана да се справим с разклоняващия коефициент. Какво става, ако фиксираме коефициента да бъде определена стойност( $k$ )?

# Решение

- При фиксиран коефициент имаме ограничение отгоре на броя върхове на дадено ниво. На второто ниво са най-много  $k$ , на третото –  $k^2$  и т.н.
- Достатъчно е да поддържаме колко върхове има на всяко ниво.
- Отново тръгваме от дърво пръчка само че в даден момент няма да можем да закачаме връх за корена, защото броя свободни върхове на първото ниво ще

# Задачи с таблици

- [https://atcoder.jp/contests/code-festival-2017-quala/tasks/code\\_festival\\_2017\\_quala\\_d](https://atcoder.jp/contests/code-festival-2017-quala/tasks/code_festival_2017_quala_d)

- Вход:

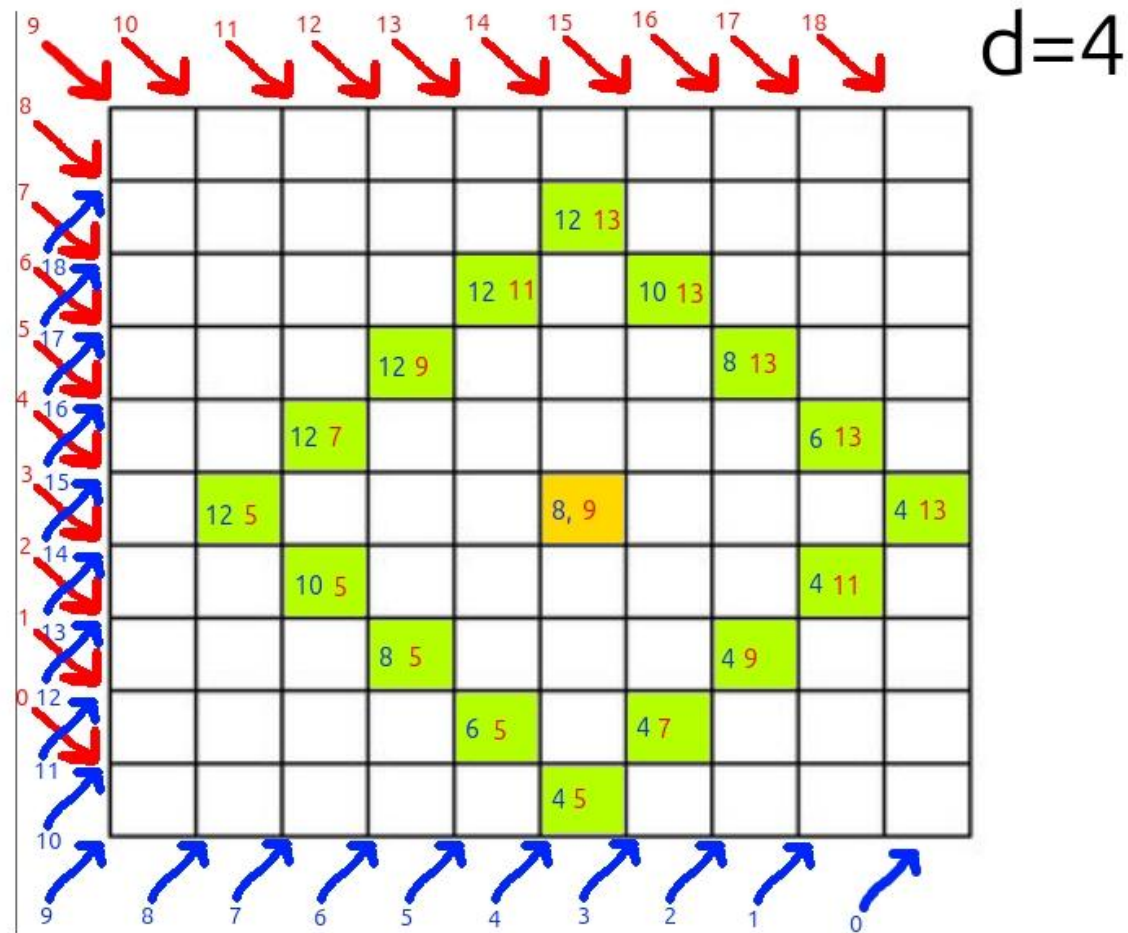
- $N, M, d$

- Изход:

- Таблица  $N \times M$ , в която всяка клетка е оцветена в един от 4 цвята така, че всеки две клетки, които са на манхатаново разстояние  $d$  един от друг, са

• В задачи за манхатаново разстояние, понякога е полезно да обърнем погледа на 45 градуса и да гледаме диагоналите.

• След като сме номерирали диагоналите, какво можем да забележим



# Решение

• Нека  $x$  и  $y$  са номерата на диагоналите, минаващи през някоя клетка.

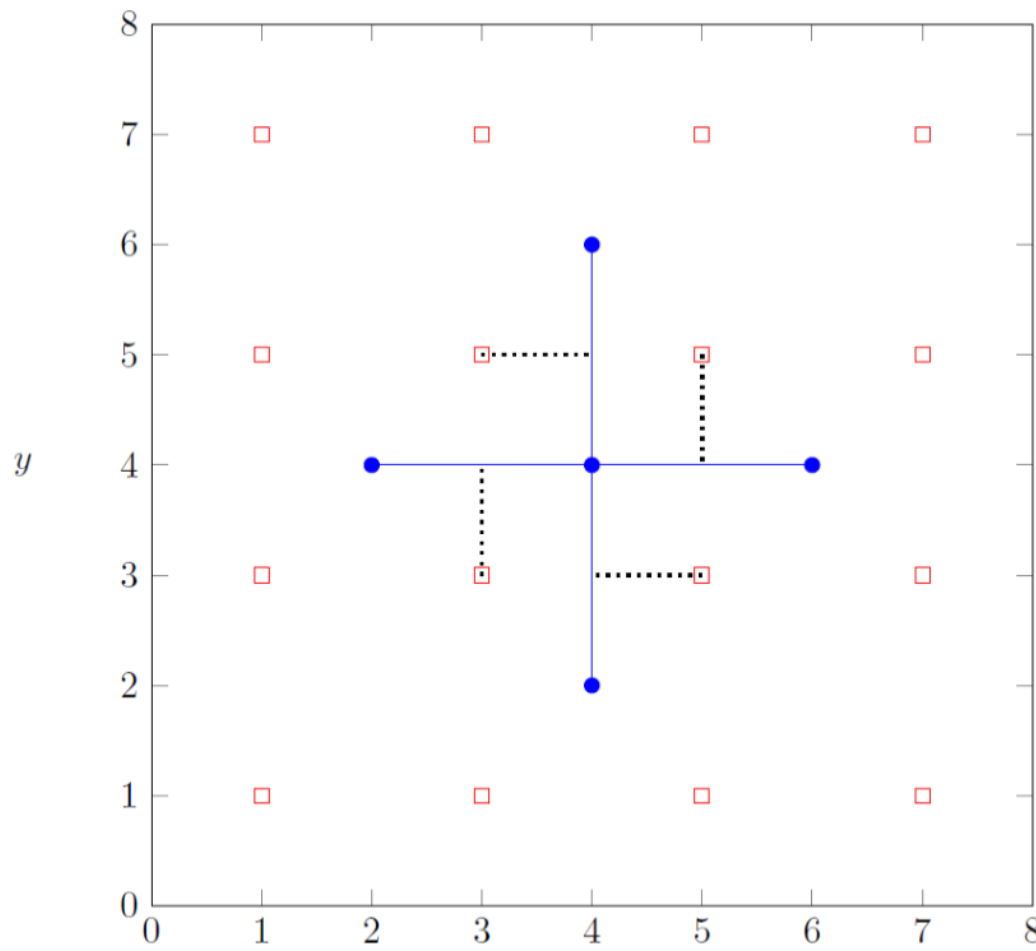
•  $\lfloor x/d \rfloor$  има различна четност от  $\lfloor (x + d) / d \rfloor$  и  $\lfloor (x - d) / d \rfloor$ . Същото и за  $y$ .

```
cin >> n >> m >> d;
char col[2][2] = {{'R', 'Y'}, {'B', 'G'}};
int offset = 2 * n * m;
for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= m; j++) {
        int ost1 = ((i + j) / d) & 1;
        int ost2 = ((i - j + offset) / d) & 1;
        cout << col[ost1][ost2];
    }
    cout << endl;
}
```

# IOI 2021 Fountain Parks

• Дадени са точки с цели, четни координати. Можем да слагаме хоризонтални или вертикални алеи с дължина 2 между две точки. На всяка точка с нечетни координати може да се построи пейка.

• Трябва да построим алеи



# IOI 2021 Fountain Parks

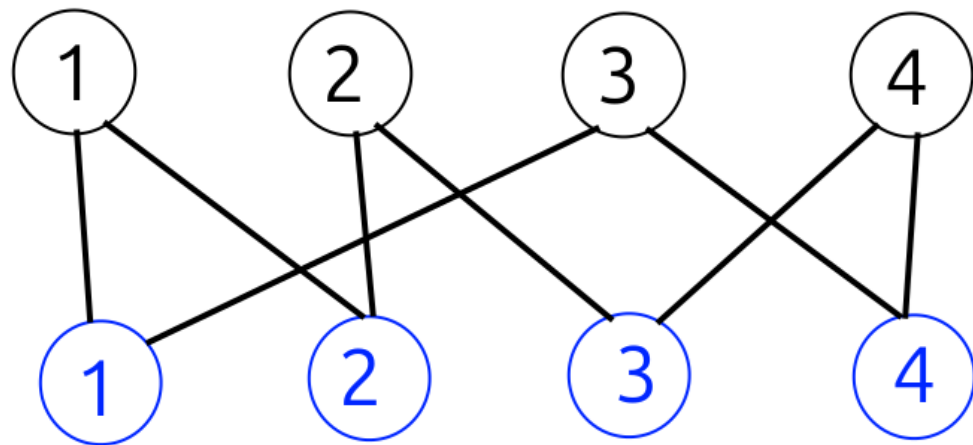
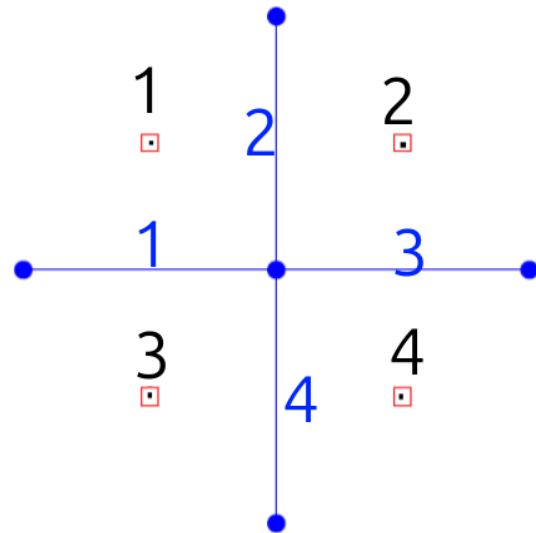
Подзадачи:

- Има най-много един начин да се поставят алеите така че графът да е свързан
- Не съществуват 4 точки, които да са ъглите на квадрат  $2 \times 2$ .

# IOI 2021 Fountain Parks

• В първата подзадача  
графът от алеите, които  
можем да построим, е или  
несвързн, или е дърво.

• Остава на всяка алея да  
съпоставим пейка, което е  
класическата задача за  
matching (решава се със  
сложност  $O(n\sqrt{n})$ ).



# IOI 2021 Fountain Parks

- За всяка алея могат да отговарят 2 пейки.
- Когато няма  $2 \times 2$  квадратчета, за колко алеи най-много може да отговаря една пейка?

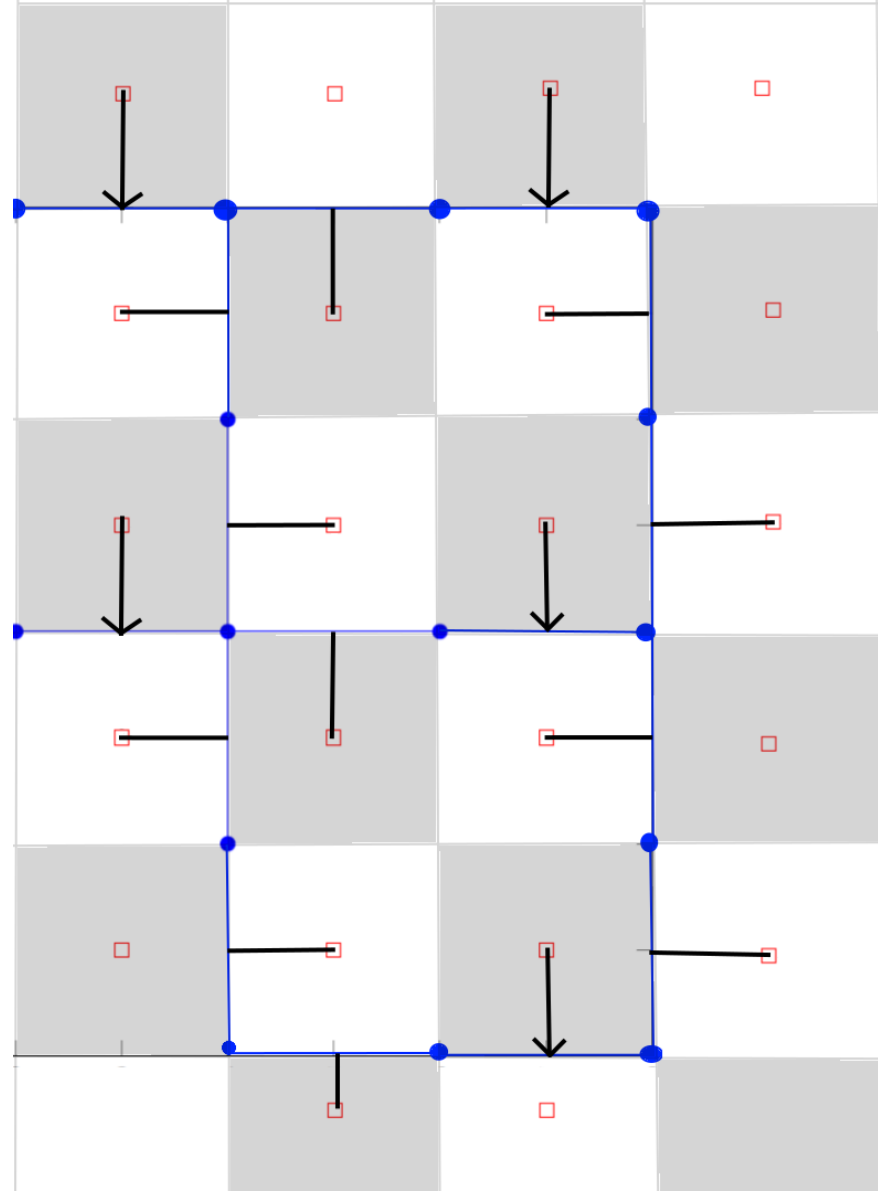
• Оцветяваме 2x2

квадратчетата шахматно.

• Черните пейки отговарят за  
хоризонтални алеи, а белите  
— за вертикални.

• За всяка алея отговарят 2  
пейки с различни цветове,  
тоест всяка алея може да  
бъде покрита.

• От друга страна, всяка пейка  
може да отговаря за



# IOI 2021 Fountain Parks

- Продължаваме с идеята от предишната подзадача, но сега трябва да махнем някои алеи, за може да я приложим.
- За всяко  $2 \times 2$  квадратче, трябва да махнем по 1 ребро (в зависимост от цвета едно вертикално или едно хоризонтално).
- В задачи, в които графът е планарен понякога е удобно да разгледаме дуалния граф – такъв, в който

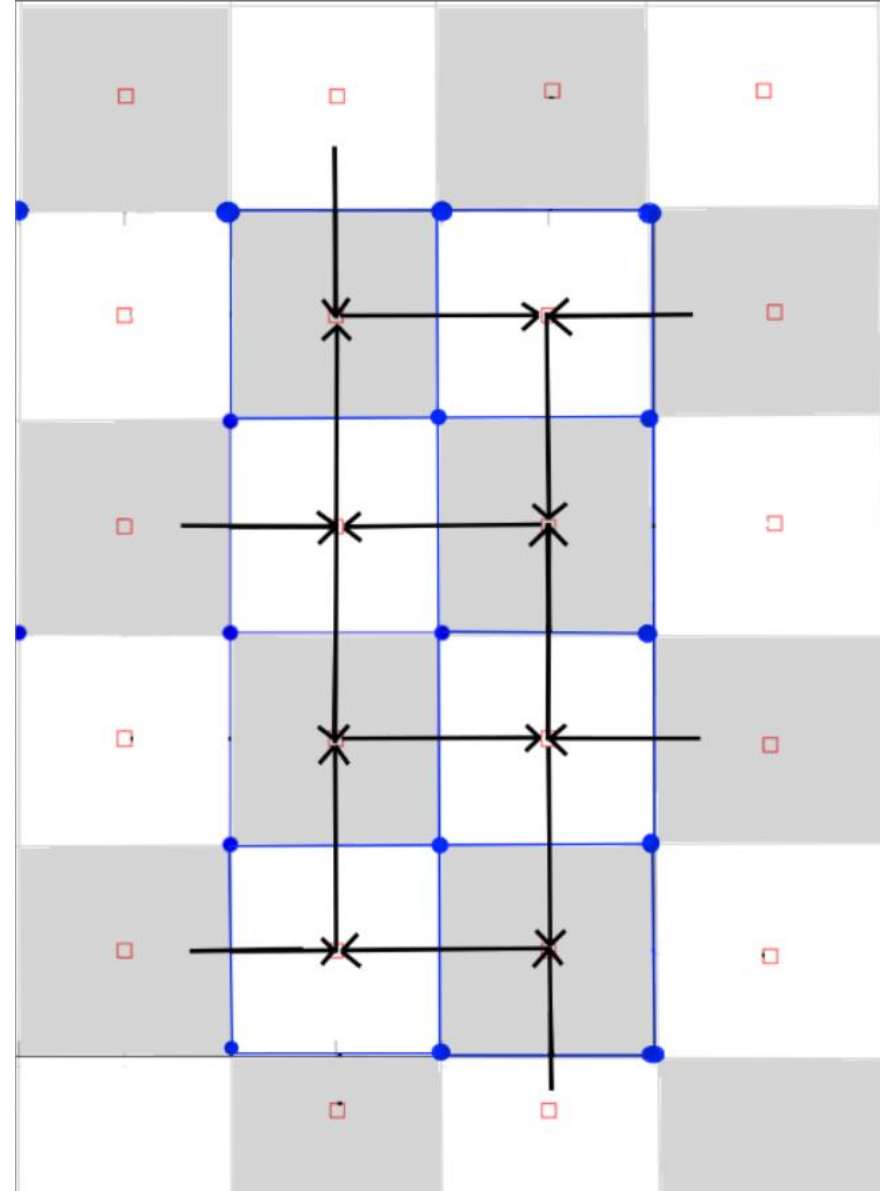
# IOI 2021 Fountain Parks

•Разглеждаме дуалния граф(граф на лицата) само за квадратчета  $2 \times 2$ . Ще имаме и връзка между квадрати  $2 \times 2$  с външното лице(външното лице ще бъде един връх в дуалния граф).

•Често в такива задачи, когато искаме да трием ребра в оригиналния граф, можем да го правим спрямо някакъв **път или дърво в дуалния**. В случая не искаме квадратите  $2 \times 2$  да си махнат и двото зони, за които отговарят, затова трябва да

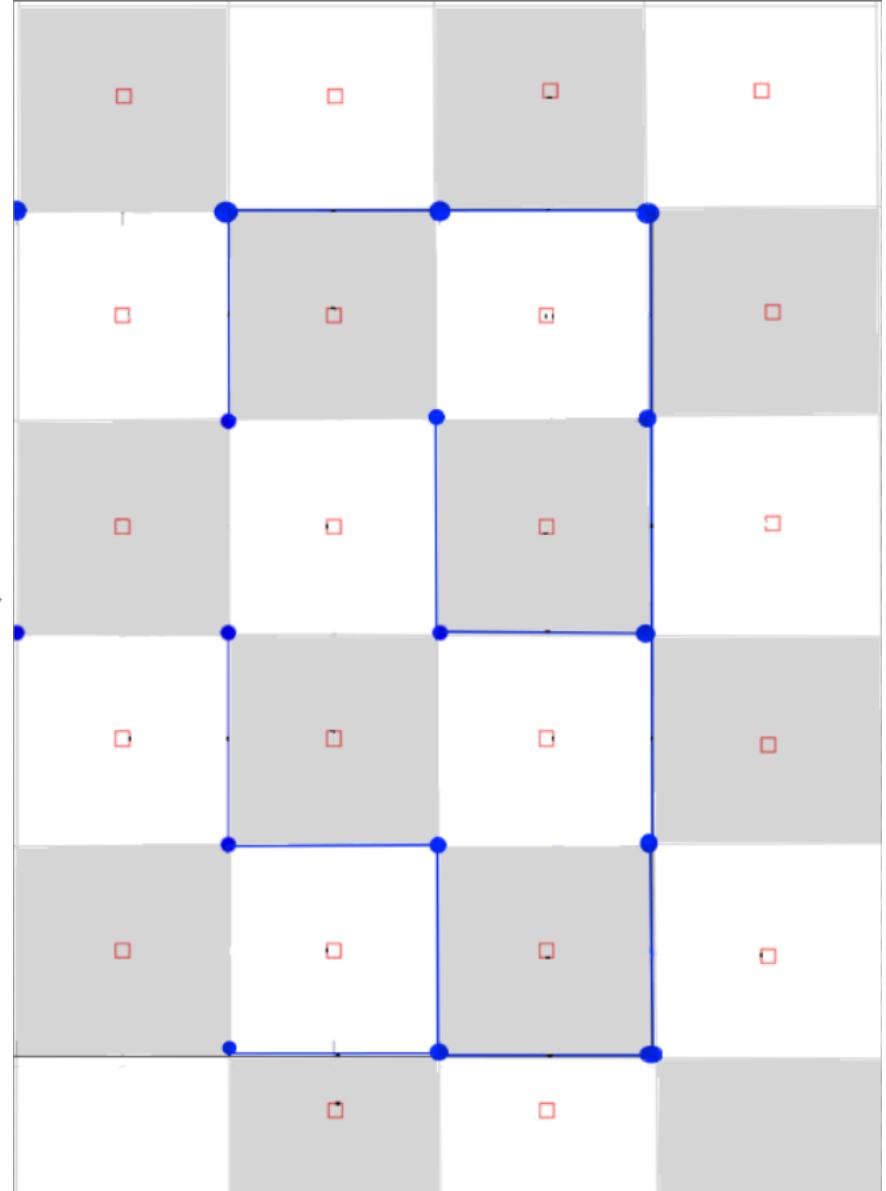
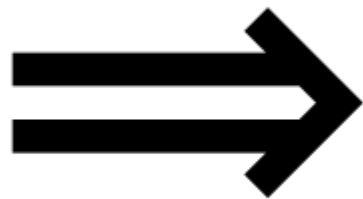
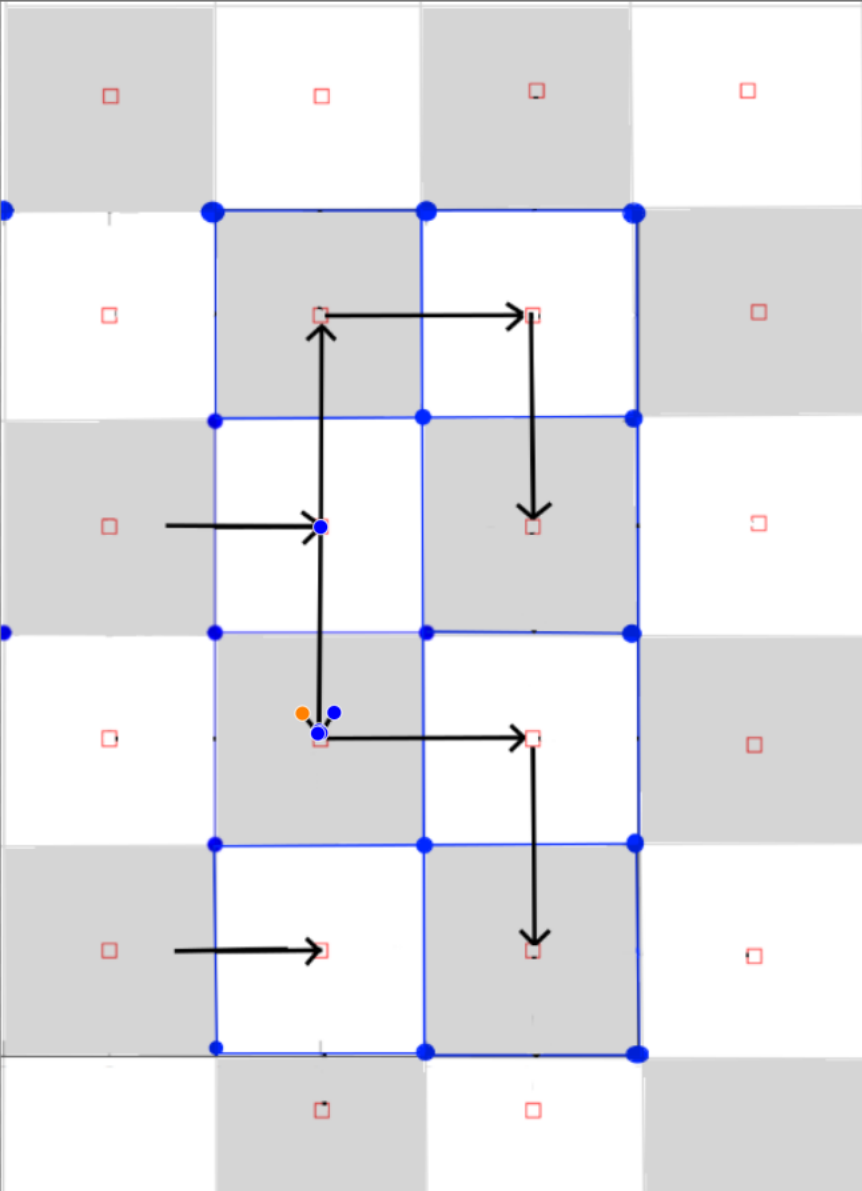
• Ще имаме **насочено** ребро от  $u$  до  $v$ , ако това ребро съответства на алея, за която отговаря връх  $v$  (например, ако реброто прекосява хоризонтална алея и  $v$  е черно квадратче, значи ще има такова ребро).

• Такива ребра ще построим и от външното лице към квадратчетата.



# IOI 2021 Fountain Parks

- Дърво с корен външното лице ще съответства на махането на една от двете алеи, за които отговаря всеки връх(квадрат) в дървото.
- Махайки тези алеи, оригиналният граф остава свързан.
- След махнето на ребрата всеки черен квадрат има точно една хоризонтална алея до него и всеки бял квадрат има точно една вертикална алея до него.

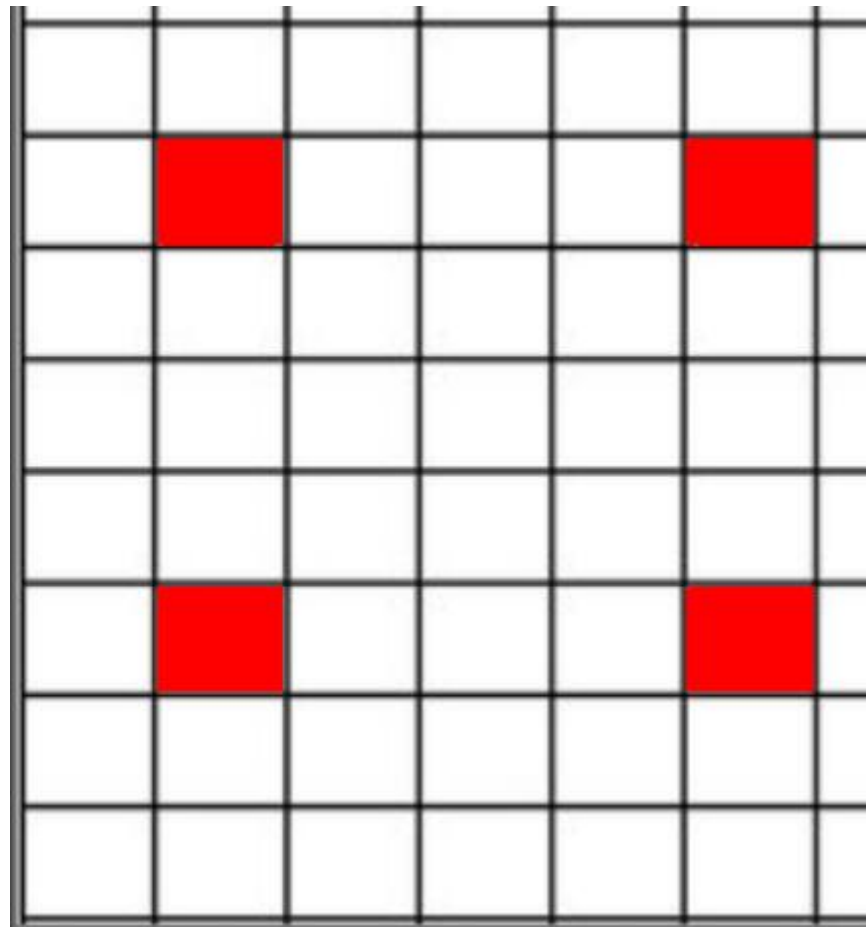


# Задачи с таблици

• [https://atcoder.jp/contests/arc226/tasks/arc226\\_c](https://atcoder.jp/contests/arc226/tasks/arc226_c)

• За  $x, y, d$  можем да оцветим клетки  $(x, y)$ ,  $(x+d, y)$ ,  $(x, y+d)$ ,  $(x+d, y+d)$  в един цвят. Един цвят може да бъде използван за точно 4 клетки.

• Търси се оцветяване на таблица  $N \times M$  с максимален



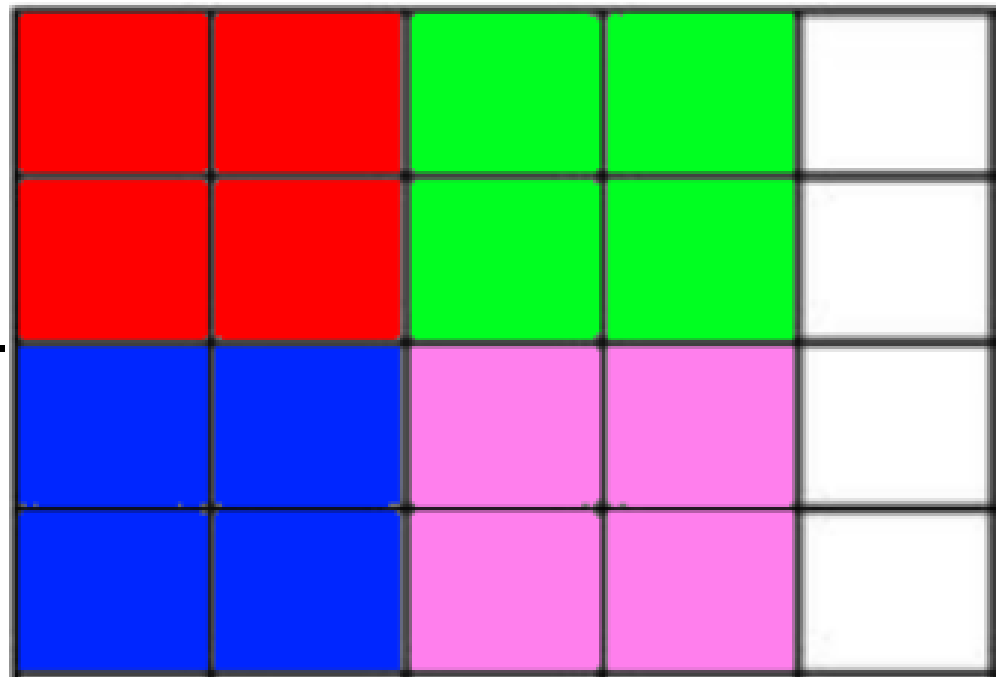
# Наблюдения

• Да намерим поне колко клетки задължително ще останат свободни. Какви случаи трябва да разгледаме?

# Наблюдения

•  $N$  и  $W$  са четни – можем да покрием цялата таблица с квадрати  $2 \times 2$ .

• Едно от  $N$  и  $W$  е нечетно – без ограничение на общността, нека  $W$  е нечетно. На всеки ред трябва да остане поне 1 свободно квадратно

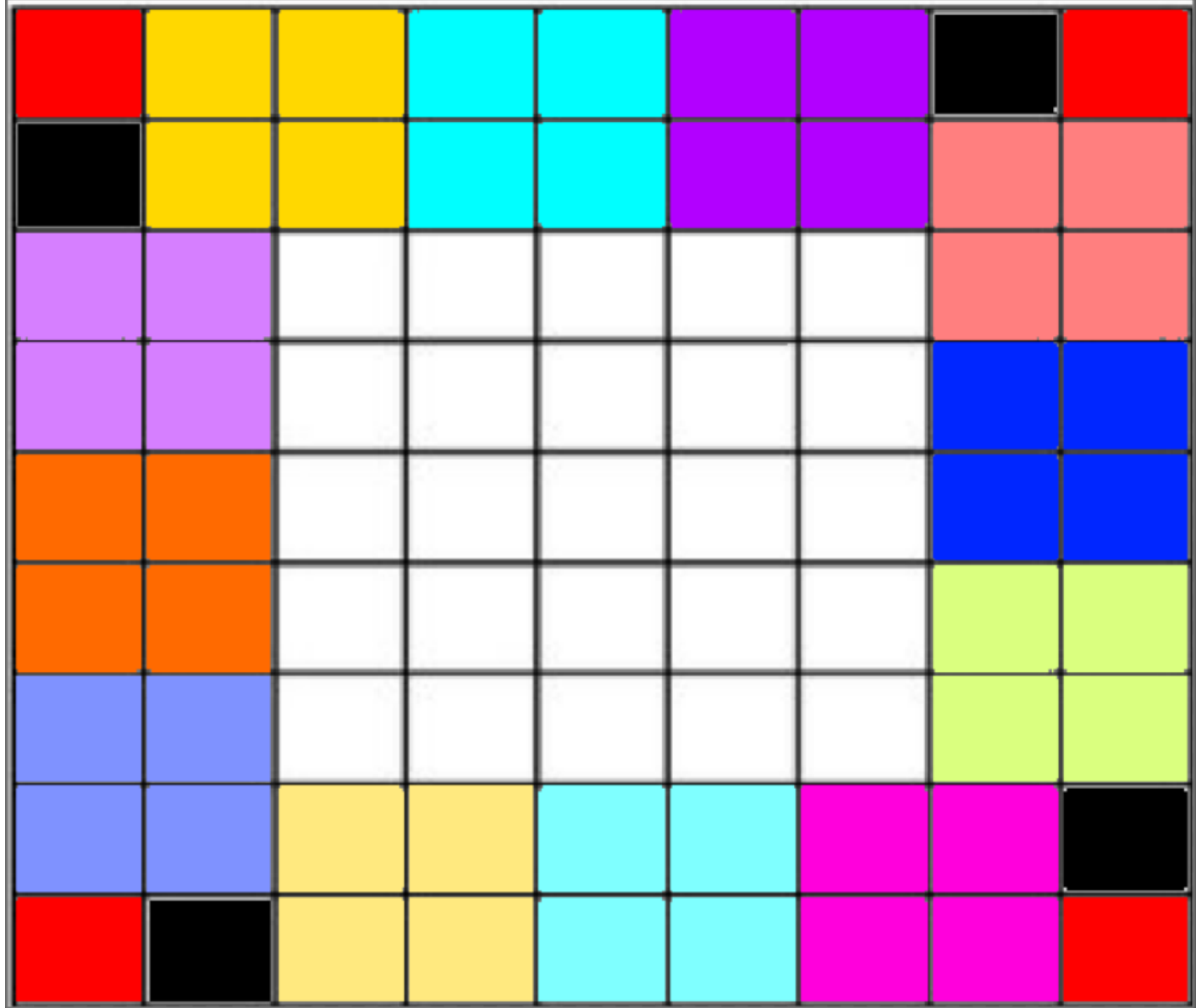


# Наблюдения

•  $H$  и  $W$  са нечетни – На всеки ред и всяка колона трябва да има поне една свободна клетка. Тоест трябва поне  $\max(H, W)$  клетки да са свободни. Оттук нататък нека  $H \geq W$ .

• Нека помислим върху по-лесен случай:  $H=W$ .  
Можете ли да измислите рекурсивна конструкция, която да свежда задачата до по-малка и същевременно спомага за това да няма повече от една свободна клетка в ред или колона?

• Намаляваме  
размера на  
таблицата на  $(N-4) \times (N-4)$  и  
използваме 4  
свободни  
клетки.

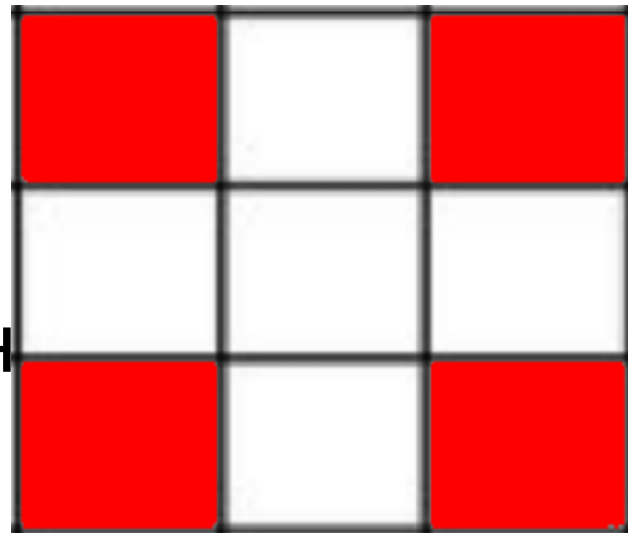


# Базови случаи

•  $N = W = 1$  – няма как да оцветим нито една клетка. От рекурсивната конструкция следва, че ще имаме точно  $N$  свободни клетки.

•  $N = W = 3$  – оказва се, че е оптимално да оставим  $(N + 2)$  свободни клетки.

Въпреки че  $N$  е долна граница за техния брой, това не е проблем(защо?)



# Когато $H \neq W$

• Можем да сложим отгоре квадрат  $W \times W$ , който да го оцветим по алгоритъма по-горе.

• Отдолу остава правоъгълник  $(H-W) \times W$ .  $(H-W)$  е четно и за този оставащ правоъгълник можем да приложим стратегията с квадратите  $2 \times 2$ .



# Когато $H \neq W$

• В зависимост от това дали при квадрата в базата стигнем до дължина 1 или 3, от квадрата ще останат  $W$  или  $W+2$  свободни клетки.

• Общо остават максимум  $H - W + W + 2 = H + 2$  свободни клетки.

• Няма как да използваме повече



# IOI 2025 – World Map

По даден граф с  $N$  върха да се конструира таблица, в която всяка клетка стои число от 1 до  $N$ , за всеки две съседни по страна клетки съществува ребро с техните номера и за всяко ребро съществува двойка съседни клетки с тези

$$N = 3, M = 2$$

$$(1, 2), (2, 3)$$

|   |   |   |
|---|---|---|
| 2 | 3 | 3 |
| 2 | 3 | 2 |
| 1 | 2 | 1 |

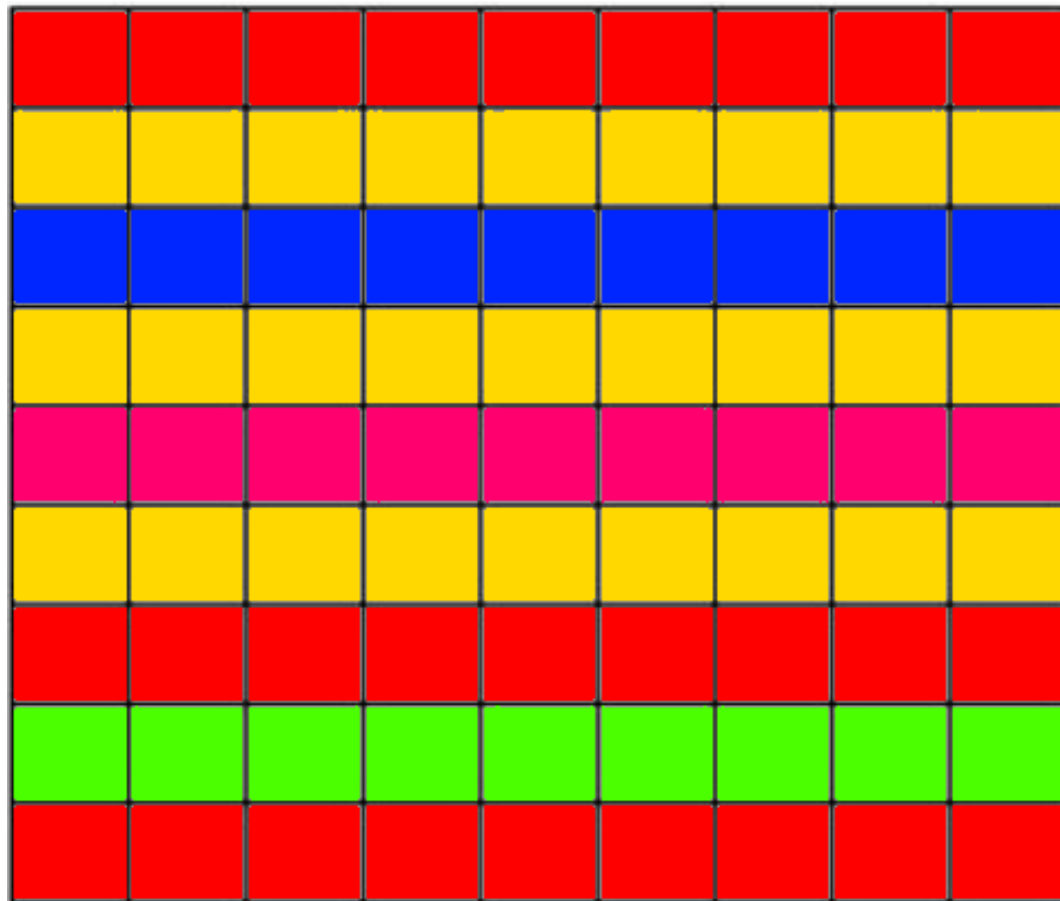
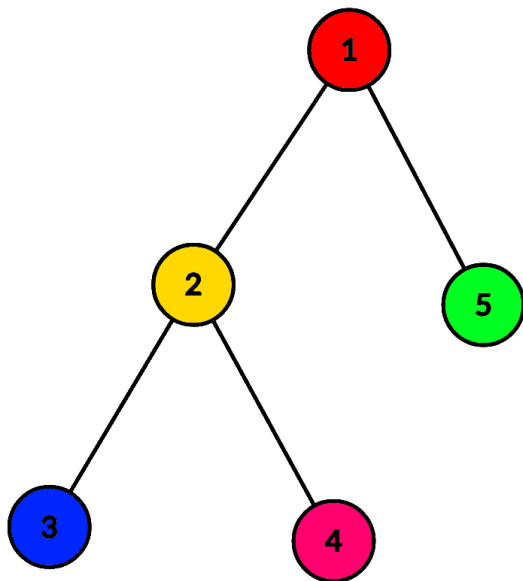
# IOI 2025 – World Map

• Можете ли да измислите лесна конструкция в случая, когато  $M = N - 1$ , която да се побира в  $2N \times 2N$  таблица?

• В общия случай, как можем да обходим списъка на съседите на всеки връх? В какъв ред да минем през всеки връх?

# IOI 2025 – World Map

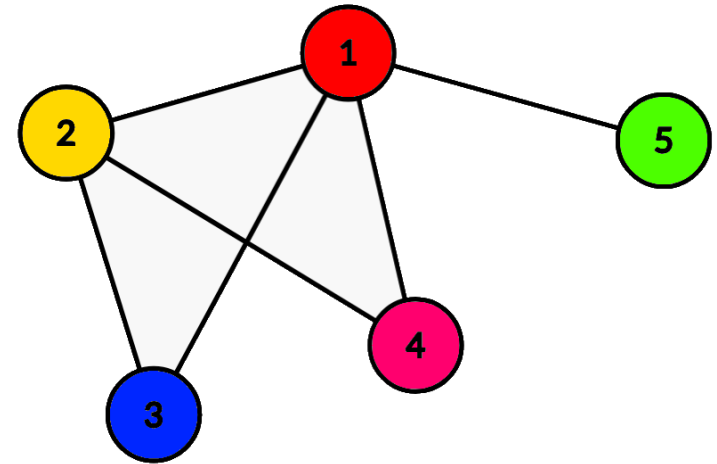
Ако графът не е свързан, очевидно е невъзможно. Тогава при

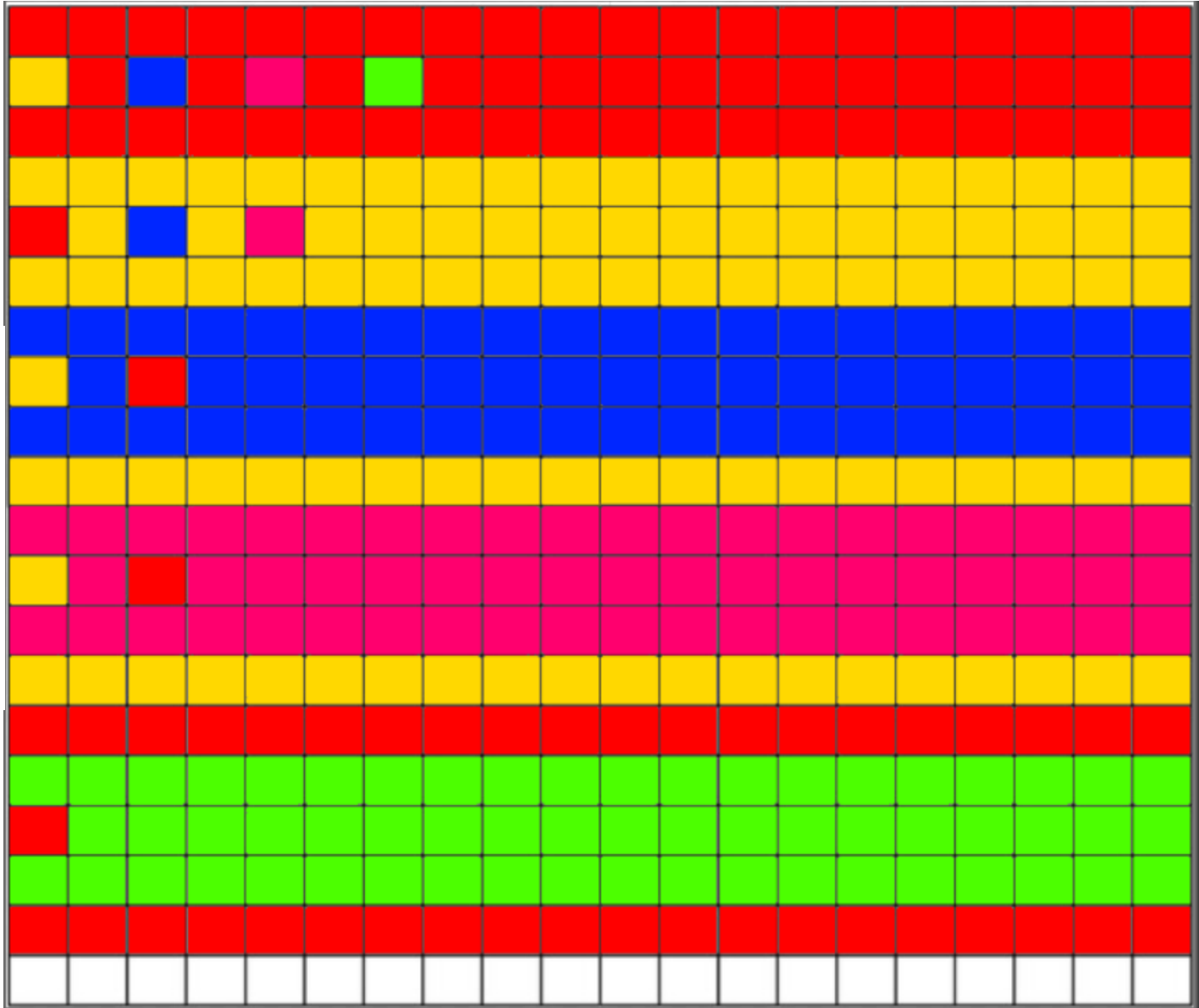
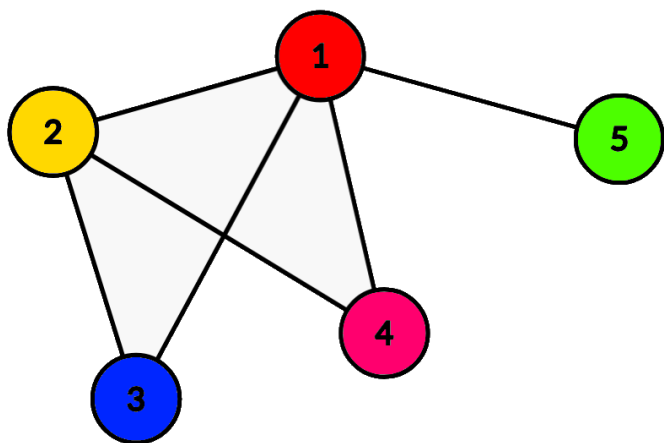


# IOI 2025 – World Map

.Обхождане на списъка на съседите с поне  $2 * (N-1)$  клетки в реда.

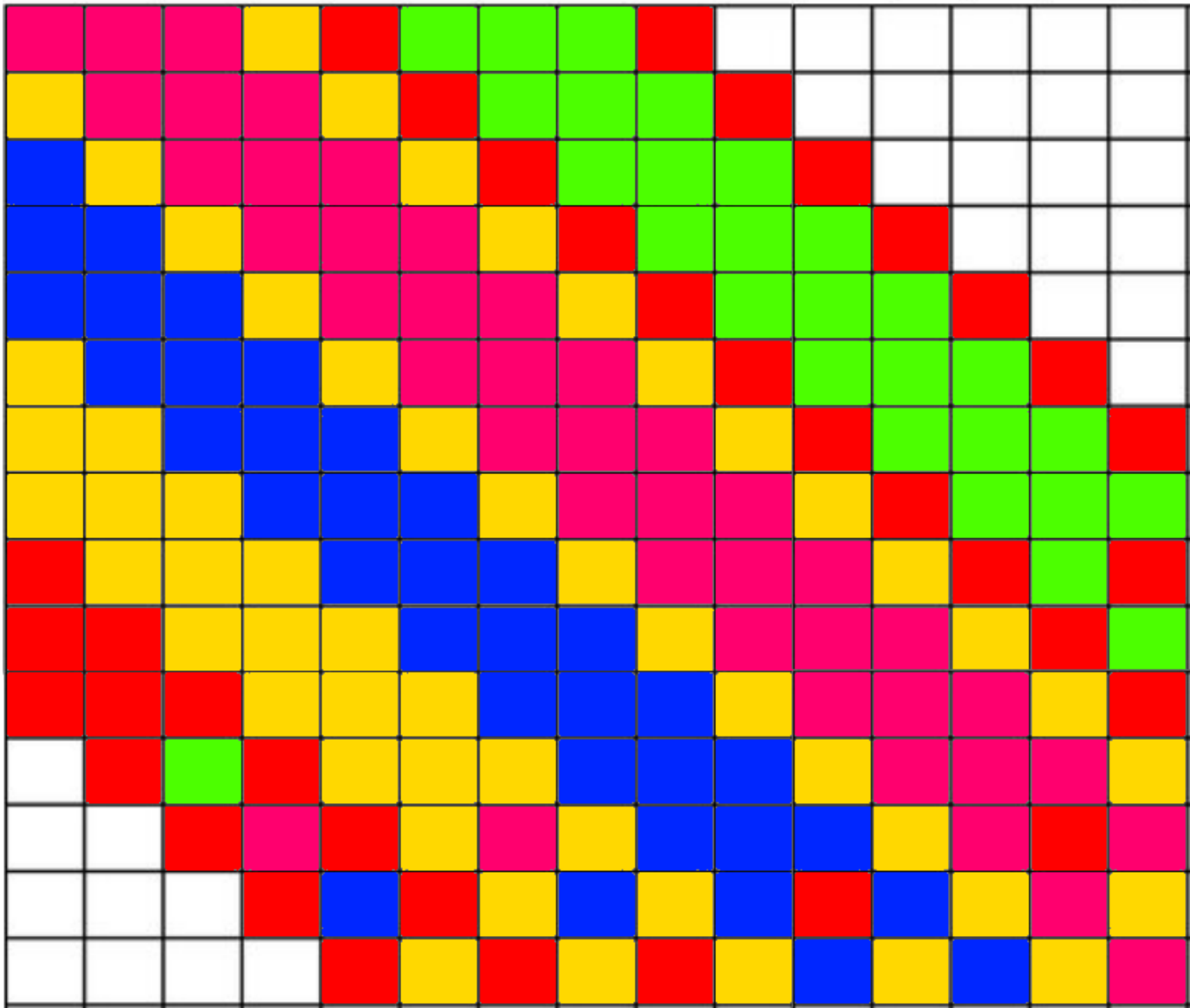
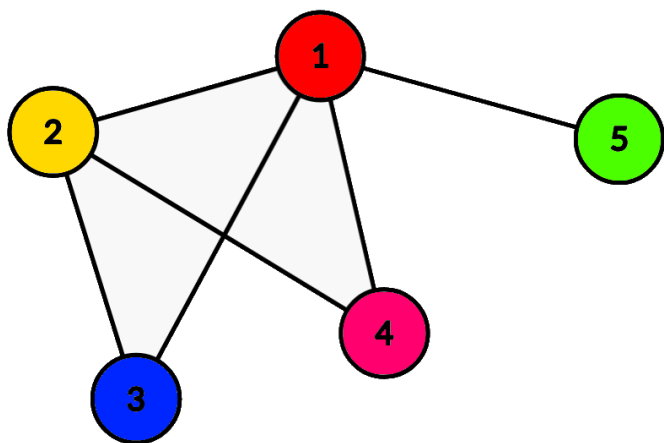
.За реда на обхождане на върховете, можем да използваме идеята от преди и да вземем покриващо дърво и на него да направим Ойлеров тур. Така за всеки връх ще





# IOI 2025 – World Map

• Можем да приложим същата идея, само че вместо редове, да използваме диагонали. Колко клетки трябва да има един диагонал, за да обходим съседите на даден връх? Какъв размер на таблицата ще ни свърши работа?



# IOI 2025 – World Map

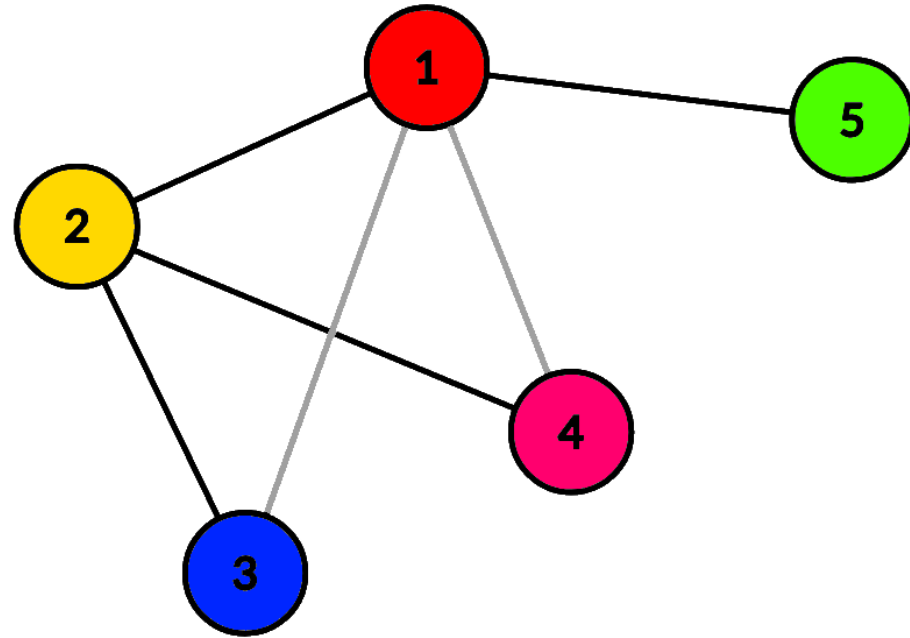
.В  $2N \times 2N$  таблица има  $4N$  диагонала, но някои от тях са много къси. Как да използваме максимално тези  $4N$  диагонала?(какво покриващо дърво да вземем и какво предимство ни дава то?)

# IOI 2025 – World Map

- DFS дърво
- Всички недървесни ребра са backedge-ове.
- Едно ребро е достатъчно да бъде разгледано само в един списък на съседите (кой от двата върха бихме избрали?)

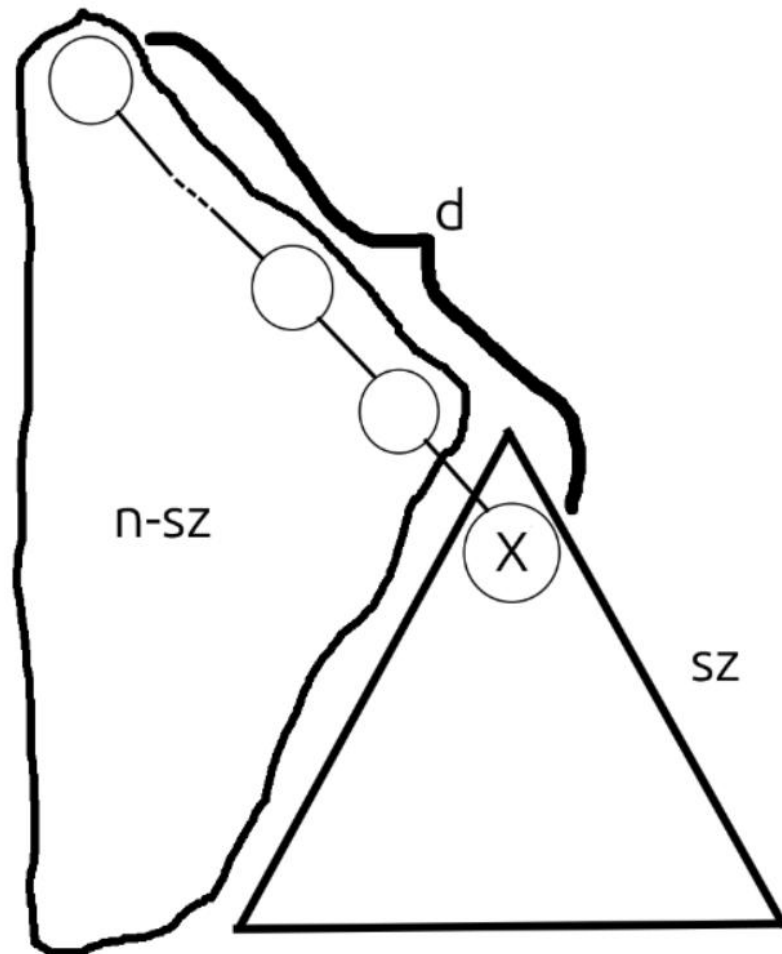
# IOI 2025 – World Map

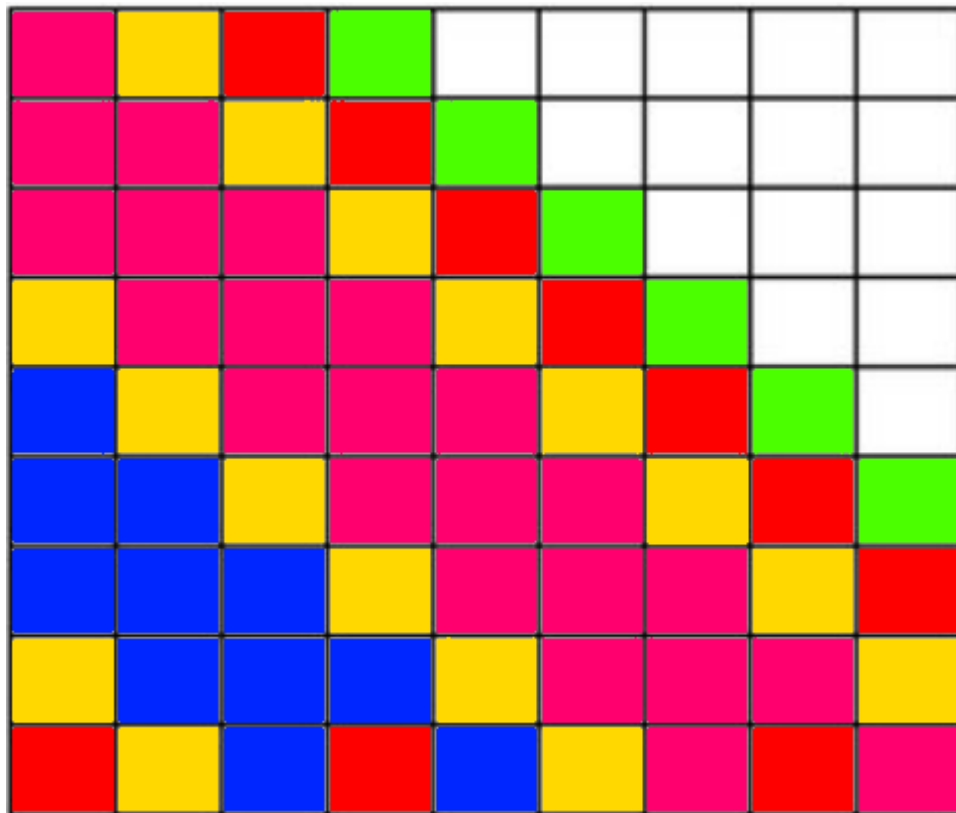
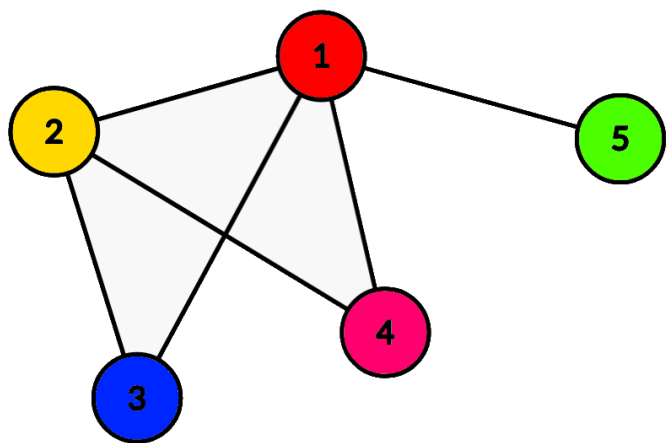
• Ако тръгваме от първия диагонал, няма да имаме пространство да разгледаме `backedge`-овете, влизащи в корена. Затова тези ребра ще ги разглеждаме, когато стигнем по-дълбоките върхове.



# IOI 2025 – World Map

Свободни диагонали:  $4N - (4(N-sz) - 2 - (d-1)) = 4N - 4N + 4sz + 2 + d - 1 = 4sz + d + 1 \geq d + 5$ . Следователно в диагонала, където изброяваме masked-edge-овете, тръгващи нагоре от  $x$ , ще има  $d+4$  клетки, което е достатъчно.





# Допълнителни задачи

- <https://codeforces.com/gym/307122>
- <https://codeforces.com/problemset/problem/906/B>
- IZHO 2019 Red Blue Table
- CEOI 2022 Drawing
- RMI 2024 Signals

**Благодаря ви за вниманието!**