

Задача АВ13. ВИДИМОСТ

 5 сек.  1024 MB

Космическият кораб USS Enterprise (със Сашка като нов капитан) пътува из вселената, която представлява безкрайна 2D мрежа от сектори. Във всеки сектор (X, Y) ¹ има маяк със сила на видимост $V_{X,Y}$. Когато е в този сектор, корабът използва маяка, за да сканира всички сектори (X', Y') с $X - V_{X,Y} \leq X' \leq X + V_{X,Y}$ и $Y - V_{X,Y} \leq Y' \leq Y + V_{X,Y}$. По този начин, видимите сектори образуват квадрат с дължина на страната $2V_{X,Y} + 1$. За всеки такъв сектор единствената видима информация е видимостта на маяка в него, т.е. $V_{X',Y'}$ (това е така, защото маяците действат едновременно като предаватели и приемници). След сканиране корабът се телепортира в един от тези видими сектори, тъй като маяците действат и като телепортатори.

За съжаление, маяците имат един критичен недостатък: те могат да обърнат оста X на сканирането или оста Y , или и двете (т.е. има 4 възможни ориентации на сканиране). Обърнете внимание, че корабът използва това евентуално обърнато сканиране, за да реши в кой сектор да се телепортира, но телепортацията също използва същата ориентация, например, ако оста Y е обърната и корабът се телепортира в лявата посока (намалявайки Y) на сканирането, той всъщност ще се телепортира вдясно. Това означава, че корабът наистина ще се телепортира в сектора, избран при сканирането.

Освен това, корабът няма памет, така че решението му къде да се телепортира трябва да зависи само от сканирането, генерирано от маяка. Освен това, стратегията на кораба трябва да бъде детерминистична – ако му е дадено едно и също сканиране, той трябва да избере да се телепортира в същия сектор в сканирането.

Корабът започва от неизвестни координати и винаги (независимо от това къде започва и независимо от ориентацията на сканирането, които получава) трябва да може да пътува в посока на увеличаване на X и Y произволно далеч (можем да кажем, че изискваме той евентуално да достигне сектор (X, Y) , такъв че $X, Y \geq 10^{100}$). За да бъде това възможно обаче, стратегията за пътуване на кораба и силите на маяците във всички сектори са от решаващо значение.

Тук се намесват Виет – трябва да проектирате правоъгълен модел P с размери $N \times M$ на видимостите от маяците, който ще се повтаря безкрайно във вселената: $V_{X,Y} = P_{X \bmod N, Y \bmod M}$. Също така трябва да проектирате стратегия за пътуване на кораба, т.е. функция, която взема предвид сканирането, генерирано от маяка, и избира към кой от тези сектори да се телепортира следващия път.

Тъй като маяците с по-висока видимост са скъпи, целта ви е да създадете валидно решение, като същевременно се опитвате да минимизирате средната видимост на маяците във Вашия модел, т.е. средната стойност на P .

Тъй като тази задача може да изглежда доста трудна, решавате да започнете с практически пример: същата задача, но в едно измерение. В тази по-проста задача изхвърляме измерението X и използваме само това на Y . По този начин Вашият модел е просто последователност с дължина M и се повтаря както следва: $V_Y = P_{Y \bmod M}$. Сканирането, генерирано от маяците, също е просто последователност с дължина $2V_Y + 1$ (съдържаща секторите Y' , такива че $Y - V_{X,Y} \leq Y' \leq Y + V_{X,Y}$).

¹ X е номерът на реда (нарастващ надолу), а Y е номерът на колоната (нарастващ надясно).

XVII INTERNATIONAL ADVANCED TOURNAMENT IN INFORMATICS ПРОЛЕТНИ СЪСТЕЗАНИЯ ПО ИНФОРМАТИКА, БУРГАС 2026

Тук има само 2 възможни ориентации за сканирането (нормална или обърната) и корабът трябва да може да достигне $Y \geq 10^{100}$.

Вашите решения за едномерния случай и двумерния случай ще бъдат оценявани отделно и може да получите точки за единия, без да имате валидно решение за другия.

Детайли по имплементацията

Трябва да имплементирате 4 функции: `getVisionPattern1d`, `getMove1d`, `getVisionPattern2d` и `getMove2d`. Първите 2 се използват за 1D случай, а вторите 2 - за 2D. Във всеки тест ще бъде извикана само една от тези двойки функции, но всичките 4 трябва да бъдат имплементирани (дори ако имплементациите за едната двойка са празни), за да бъде решението валидно (за компилиране).

```
std::vector<int> getVisionPattern1d()
```

Тази функция ще бъде извикана веднъж, ако тестът е за 1D случай. Тя трябва да върне последователността P с дължина M - моделът 1D на силата на зрението, който ще се повтаря. M и всички елементи на P трябва да са между 1 и 60.

```
int getMove1d(std::vector<int> v)
```

Тази функция ще бъде извикана веднъж за всяко възможно сканиране, ако тестът е за 1D случай. Дадено ѝ е сканиране, генерирано от маяк, като последователност с дължина $2V + 1$ (където V е силата на зрението на маяка в централния сектор в последователността). Трябва да върне в кой сектор трябва да се телепортира корабът, като индекс T в последователността, така че $0 \leq T < 2V + 1$.

```
std::vector<std::vector<int>> getVisionPattern2d()
```

Тази функция ще бъде извикана веднъж, ако тестът е за 2D случай. Тя трябва да върне правоъгълна таблица P с размер N по M - 2D моделът на силата на зрението, който ще се повтори. N , M и всички елементи на P трябва да са между 1 и 60.

```
std::pair<int, int> getMove2d(std::vector<std::vector<int>> v)
```

Тази функция ще бъде извикана веднъж на възможно сканиране, ако тестът е за 2D случай. Дадено е сканиране, генерирано от маяк, като квадратна таблица с размери $2V + 1$ на $2V + 1$ (където V е силата на видимост на маяка в централния сектор в квадрата). Трябва да върне в кой сектор трябва да се телепортира корабът, като двойка индекси S и T в таблицата, така че $0 \leq S, T < 2V + 1$.

За дадената размерност D (1 или 2) на теста, грейдерът ще провери дали Вашата програма генерира валиден шаблон - дали прави валидни избори за телепортация за всички възможни сканирания и дали Вашите решения работят независимо от началните координати и последователността на ориентациите на сканиранията.

**XVII INTERNATIONAL ADVANCED TOURNAMENT IN INFORMATICS
 ПРОЛЕТНИ СЪСТЕЗАНИЯ ПО ИНФОРМАТИКА, БУРГАС 2026**

Ограничения

- $1 \leq D \leq 2$
- $1 \leq N, M, V_{X,Y}, V_Y \leq 60$

Подзадачи и оценяване

Подзадача	Точки	D	A^*
1	24	1	1.5
2	76	2	1.125

Вашият резултат за дадена подзадача (ако решението ви за нея е правилно) зависи от средната видимост във Вашия шаблон P за нея. Нека това е A и нека целта за подзадачата бъде A^* . Вашият резултат (дял от точките за подзадачата) ще бъде:

$$1 - \left(1 - \frac{A^* - 1}{\max(A, A^*) - 1} \right)^{0.8}$$

Примерен модел

Да предположим, че Вашето решение връща следния модел с $N = 3$ и $M = 2$:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

Сега нека разгледаме възможните сканирания в сектор $(0, 1)$, който има сила на зрение 2. Има 4 възможни ориентации (някои от които водят до същото сканиране на този модел):

Ориентация 0: Без обръщания

$$\begin{bmatrix} 4 & 3 & 4 & 3 & 4 \\ 6 & 5 & 6 & 5 & 6 \\ 2 & 1 & 2 & 1 & 2 \\ 4 & 3 & 4 & 3 & 4 \\ 6 & 5 & 6 & 5 & 6 \end{bmatrix}$$

Ориентация 1: Оста Y е обърната

$$\begin{bmatrix} 4 & 3 & 4 & 3 & 4 \\ 6 & 5 & 6 & 5 & 6 \\ 2 & 1 & 2 & 1 & 2 \\ 4 & 3 & 4 & 3 & 4 \\ 6 & 5 & 6 & 5 & 6 \end{bmatrix}$$

Ориентация 2: Обърната ос X

$$\begin{bmatrix} 6 & 5 & 6 & 5 & 6 \\ 4 & 3 & 4 & 3 & 4 \\ 2 & 1 & 2 & 1 & 2 \\ 6 & 5 & 6 & 5 & 6 \\ 4 & 3 & 4 & 3 & 4 \end{bmatrix}$$

Ориентация 3: Обърнати са и двете оси

$$\begin{bmatrix} 6 & 5 & 6 & 5 & 6 \\ 4 & 3 & 4 & 3 & 4 \\ 2 & 1 & 2 & 1 & 2 \\ 6 & 5 & 6 & 5 & 6 \\ 4 & 3 & 4 & 3 & 4 \end{bmatrix}$$

Тъй като ориентации 0 и 1 са идентични и корабът е детерминистичен, за тях ще бъде направено само едно извикване на `getMove2d`. Да предположим, че Вашето

XVII INTERNATIONAL ADVANCED TOURNAMENT IN INFORMATICS ПРОЛЕТНИ СЪСТЕЗАНИЯ ПО ИНФОРМАТИКА, БУРГАС 2026

решение връща преместването $(0, 1)$ при това сканиране. При ориентация 0 това би означавало преместване с 1 наляво и 2 нагоре, докато при ориентация 1 – преместване с 1 надясно и 2 нагоре.

Формат на грейдъра

Първо се въвежда D . След това грейдъра ще извърши всички извиквания на Вашата функция: първо ще получи шаблона P и след това ще кешира всички избори за телепортация. След това ще започне взаимодействие с конзолата, като поиска началните координати на кораба. След това ще отпечата текущото състояние: координати и “сурово” (без обръщания) сканиране. След това ще поиска да се използва ориентацията (0 за без обръщания, 1 за обръщане на оста Y , 2 за обръщане на оста X и 3 за обръщане и на двете). След това ще се отпечата сканирането в дадената ориентация, както и избраното движение на кораба. Корабът ще бъде преместен и това ще се повтори. Обърнете внимание, че примерният грейдър няма автоматично да провери дали решението ви е правилно.