

Moving Robots

Adott egy 8×8 -as sakktábla, amelynek minden mezőjén egy robot áll. Mindegyik robot egymástól függetlenül k lépést tesz, és több robot is állhat ugyanazon a mezőn.

Minden lépés során egy robot egy lépést tesz balra, jobbra, felfelé vagy lefelé, de nem léphet ki a tábláról. A robot véletlenszerűen választ egy irányt azok közül, amerre léphet.

A feladat az, hogy kiszámítsd a várható üres mezők számát k lépés után.

Bemenet

Az egyetlen sor egy egész számot tartalmaz: k .

Kimenet

Nyomtasd ki a várható üres mezők számát hat tizedesjegyre kerekítve (fél esetén párosra kerekítve).

Korlátok

$$1 \leq k \leq 100$$

Példa

Bemenet:

10

Kimenet:

23.120740

Probléma elemzése

A sakktábla 8 x 8-as méretű, vagyis 64 mezőből áll, és kezdetben minden mezőn áll egy-egy robot. A robotok véletlenszerűen mozognak egy lépésben balra, jobbra, felfelé vagy lefelé úgy, hogy nem lépnek le a tábláról. A cél a várható üres mezők számának kiszámítása k lépés után.

Algoritmus kidolgozása

1. Valószínűségi modellezés: Minden robot négy irányba mozoghat, de csak akkor, ha az adott irány nem vezet le a tábláról. Egy adott mezőre annak a valószínűsége kell, hogy kiszámoljuk, hogy ott nincs robot k lépés után.
2. Dinamikus programozás (DP): Létrehozunk egy DP-táblát, amely tárolja a valószínűségeket arra vonatkozóan, hogy egy adott mezőn robot van-e k lépés után. Kezdetben minden mezőn 100% valószínűséggel van robot. Lépésről lépésre frissítjük a mezők valószínűségeit a robotok mozgásának iránya alapján.
3. Iteráció a lépések felett: Minden mező esetében a valószínűséget átlagoljuk azokból a mezőkből, amelyekből a robot elérhette azt a mezőt.

Magyarázat:

1. DP táblázat: Egy 3D táblázatot használunk, ahol $dp[k][i][j]$ tárolja annak a valószínűségét, hogy k lépés után az (i, j) -edik mezőn van robot.
2. Szomszédos mezők számítása: Minden lépés során meghatározzuk, hogy az adott mező szomszédos mezőire milyen arányban kerülhet át a valószínűség. Ha például egy mezőnek 4 szomszédja van, akkor mindegyik szomszéd $1/4$ -ed valószínűséget kap.
3. Üres mezők kiszámítása: Az $1 - dp[k][i][j]$ érték adja meg annak a valószínűségét, hogy az (i, j) -edik mező üres k lépés után. Ezeket az értékeket összeadjuk minden mezőre, hogy megkapjuk a várható üres mezők számát.

Példa futtatás:

Bemenet: 10

Kimenet: 23.120740

Hatékonyság

Időbonyolultság: $O(K * N^2)$, mivel K lépést kell szimulálni, és minden lépésben $N \times N$ mezőt kell frissíteni.

Térbonyolultság: $O(K * N^2)$, mivel a DP táblázat K -hoz és N -hez kötött.

CSES teszt eredmények

CSES Problem Set

Moving Robots

[TASK](#) | [SUBMIT](#) | [RESULTS](#) | [STATISTICS](#) | [TESTS](#) | [QUEUE](#)

Submission details

Task:	Moving Robots
Sender:	pipdom
Submission time:	2024-12-15 10:04:42 +0200
Language:	Python3 (PyPy3)
Status:	READY
Result:	ACCEPTED

Test results ▲

test	verdict	time	
#1	ACCEPTED	0.07 s	»»
#2	ACCEPTED	0.08 s	»»
#3	ACCEPTED	0.09 s	»»
#4	ACCEPTED	0.09 s	»»
#5	ACCEPTED	0.09 s	»»
#6	ACCEPTED	0.12 s	»»
#7	ACCEPTED	0.12 s	»»
#8	ACCEPTED	0.13 s	»»
#9	ACCEPTED	0.13 s	»»
#10	ACCEPTED	0.13 s	»»
#11	ACCEPTED	0.07 s	»»
#12	ACCEPTED	0.09 s	»»
#13	ACCEPTED	0.09 s	»»
#14	ACCEPTED	0.08 s	»»
#15	ACCEPTED	0.14 s	»»
#16	ACCEPTED	0.09 s	»»