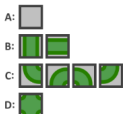


Lausn á *Pipe Rotation*

Bergur Snorrason

30. janúar 2023

- ▶ Okkur er gefið $n \times m$ ($1 \leq n, m \leq 100$) borð af spilum af eftirfarandi gerðum:



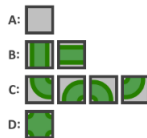
- ▶ Við eigum að ákvarða hvort hægt sé að snúa spilunum þannig að allar hliðar sem eru grænar eru gagnstæðar hliðum sem eru einnig grænar.
- ▶ Takið eftir að við megum ekki endurraða spilunum og að hliðar sem snúa út mega ekki vera grænar.

- ▶ Sjáum dæmi.



- ▶ Efra dæmið er ekki löglegt en neðra er löglegt.

- ▶ Sjáum fyrst að ef við vitum hvað er vinstra megin og fyrir ofan spil þá vitum við hvernig það þarf að snúa.
- ▶ Ef hliðin fyrir ofan er auð og hliðin til vinstri er auð þá þarf spilið að vera af gerð A eða af gerð C í öðrum snúning.
- ▶ Ef hliðin fyrir ofan er auð og hliðin til vinstri er græn þá þarf spilið að vera af gerð B í seinni snúning eða af gerð C í þriðja snúning.
- ▶ Ef hliðin fyrir ofan er græn og hliðin til vinstri er auð þá þarf spilið að vera af gerð B í fyrri snúning eða af gerð C í fyrsta snúning.
- ▶ Ef hliðin fyrir ofan er græn og hliðin til vinstri er græn þá þarf spilið að vera af gerð C í fjórða snúning eða af gerð D.



- ▶ Nú getum við labbað í gegnum öll spilin og ákvarðað hvernig hvert spil snýr ef við byrjum á efstu línunni og vinnum okkur niður, og skoðum hverja línu frá vinstri til hægri.
- ▶ Það er þínu vinna að útfæra þetta útaf öllum tilfellunum sem þarf að hafa í huga.

```
4 void imp()  
5 {  
6     printf("Impossible\n");  
7     exit(0);  
8 }
```

```

10 int main()
11 {
12     int i, j, n, m;
13     scanf("%d%d", &n, &m);
14     char s[n][m + 1], a[n + 1][m + 1], b[n + 1][m + 1];
15     for (i = 0; i < n + 1; i++) for (j = 0; j < m + 1; j++)
16         a[i][j] = b[i][j] = 0;
17     for (i = 0; i < n; i++) scanf("%s", s[i]);
18     for (i = 0; i < n; i++) for (j = 0; j < m; j++)
19     {
20         if (a[i][j] == 0 && b[i][j] == 0)
21         {
22             if (s[i][j] == 'C') a[i + 1][j]++, b[i][j + 1]++;
23             else if (s[i][j] != 'A') imp();
24         }
25         else if (a[i][j] == 1 && b[i][j] == 0)
26         {
27             if (s[i][j] == 'B') a[i + 1][j]++;
28             else if (s[i][j] == 'C') b[i][j + 1]++;
29             else imp();
30         }
31         else if (a[i][j] == 0 && b[i][j] == 1)
32         {
33             if (s[i][j] == 'B') b[i][j + 1]++;
34             else if (s[i][j] == 'C') a[i + 1][j]++;
35             else imp();
36         }
37         else
38         {
39             if (s[i][j] == 'D') a[i + 1][j]++, b[i][j + 1]++;
40             else if (s[i][j] != 'C') imp();
41         }
42     }
43     for (i = 0; i < n; i++) if (b[i][m]) imp();
44     for (i = 0; i < m; i++) if (a[n][i]) imp();
45     printf("Possible\n");
46     return 0;
47 }

```

- ▶ Þessi lausn er lítið annað en tvöföld for-lykkja sú ytri af lengd n og sú innri af lengd m .
- ▶ Svo tímaflækjan er $\mathcal{O}(n \cdot m)$.

