

Wzrost Siempińskiego

32 x 65

Jedynymowym autanet kombikany działający na tablicy $t[x]$.
 W każdym wierszu będzie jedna itonaga po czasie $t=0, t < \dots, t+r$

Zachowanie dżadu dreslajz repity:

- 0 < 0 > 0 → 0 (1)
- 0 < 0 > 1 → 1 (2)
- 0 < 1 > 0 → 0 (3)
- 0 < 1 > 1 → 0 (4)
- 1 < 0 > 0 → 1 (5)
- 1 < 0 > 1 → 0 (6)
- 1 < 1 > 0 → 0 (7)
- 1 < 1 > 1 → 0 (8)

0 < 0 > 1 → 1 oznacza że
 jeśli $t=0$ jest w stanie < 0 >
 i po lewej stronie jest 0,
 a po prawej 1, to dla
 $t=a+1$ przejść do stanu 1.
 itd.

Wzrost dla $x=1; x < 32; x+r$
 (zaczynamy od 1, aby „przejść” w lewo
 bez wyjścia poza zakres)

Wartość początkowa

$t=0 \rightarrow x=00\dots010\dots00$

znaczenie pozycji = 1, więc

int tab[32][65] = {0};

tab[0][32] = 1;

if (tab[t+1][x-1] == 0 & tab[t][x] == 1
 & tab[t+1][x+1] == 1)

tab[t+1][x] = 1; itd.




```
#include <stdio.h>
```

```
#define HEIGHT 32
```

```
#define WIDTH 65
```

```
void printing-triangle(char *sierpinski)
```

```
int main()
```

```
{
```

```
char sierpinski[HEIGHT][WIDTH] = {0};
```

```
sierpinski[0][32] = 1;
```

```
int t, x;
```

```
for (t=0; t < HEIGHT; t++)
```

```
{  
    for (x=1; x < WIDTH; x++)
```

```
{  
    if (sierpinski[t][x-1] == 0 && sierpinski[t][x] == 0 && sierpinski[t][x+1] == 0  
        sierpinski[t][x] sierpinski[t+1][x] = 0;
```

```
else if (sierpinski[t][x-1] ...)
```

```
    sierpinski[t+1][x] = 1;
```

```
...
```

```
}
```

```
}
```

```
printing-triangle(sierpinski);
```

```
return 0;
```

```
}
```

↑ pointer ułatwi szukanie
i porównanie się po obu
indeksach jednocześnie

funkcja `printing-triangle()` wypisuje w każdej iteracji dany znak

jeżeli natrafimy na pozycję z wartością $= 0$, wypisuje spacje
jeżeli natrafimy na 1 wypisuje znak $*$, który będzie tworzył trójkąt

piąte $*$ wypisuje nam się dopiero w potencie, bo dopiero w potencie tablicy dla $t=0$ natrafiamy na $x=1$

`[sierpinski[0][32]=1;`

`sierpinski[32][65]=10%;` \rightarrow

potem kolejne dwa $*$ drukując,
bo w kolejnej iteracji tym
samym wzoru matematycznym
na $|x+1|=1$ i $|x-1|=1$

itd.

```
void printing-triangle(char *sierpinski)
```

```
{
```

```
    int t, x;
```

```
    for (t=0; t<32; t++)
```

```
    {
```

```
        if (sierpinski[t]==1)
```

```
            printf("*");
```

```
        else printf(" ");
```

```
    }
```

```
    sierpinski++;
```

```
    printf("\n");
```