6.3 Online Search: Searching for a Point on a Line

- Suppose cow needs to find some distinguished point ($\hat{=}$ hay) on a line.
Assume n steps away along the line.
If the cow knows that the point is to its left (or to its right), whether or not it knows the actual distance
→ can optimally find it in n steps
If cow does know n steps away, but not left / right
→ easy to show optimal: Go n steps left, turn, go $2n$ right
Suppose: cow does not know how far away the point is
↳ Min # steps as fct. of n?

Any algorithm can be described as function f

- $f(i) = \#$ steps to $\begin{cases} \text{left} \\ \text{right} \end{cases}$ before the i-th turn, odd ~~steps~~ terms - left, even ~~steps~~ - right

→ $f(1)$ steps to left
$f(2)$ — " — right
⋮

Necessary: $f(i) \geq f(i-2) + 1 \quad \forall i \geq 1 \quad (f(-1) = f(0) = 0)$

Def. 6.18 (Linear Spiral Search): $f(i) = 2^i \quad \forall i \geq 1$

Theorem 6.19: Linear Spiral Search (LSS) is 9-competitive.

Proof:
Worst-case: point is short after the turn location of LSS

$\boxed{\text{II}}$

$\Rightarrow$ OPT $= 2^i + \varepsilon$

$$LSS = 2 \cdot \sum_{k=1}^{i+1} 2^k + 2^i + \varepsilon$$

$$= 2 \cdot \sum_{k=0}^{i+1} 2^k - 1 + 2^i + \varepsilon$$

$$= 2(2^{i+2} - 1) - 1 + 2^i + \varepsilon$$

$$= 2 \cdot 2^{i+2} + 2^i - 3 + \varepsilon$$

$$= 2^i (2 \cdot 2^2 + 1) - 3 + \varepsilon$$

$$= 9 \cdot 2^i - 3 + \varepsilon$$

$$\Rightarrow \frac{LSS}{OPT} = \frac{9 \cdot 2^i - 3 + \varepsilon}{2^i + \varepsilon} \to 9$$