

動的計画法 (Frog 問題を例に)

動的計画法 (Dynamic Programming)

→ 与えられた問題全体を一連の部分問題に上手に分解し、各部分問題に対する解を メモ化 (記憶) する。そして部分問題から、より大きな部分問題へと順に解を求めていく手法。

例) Frog 問題

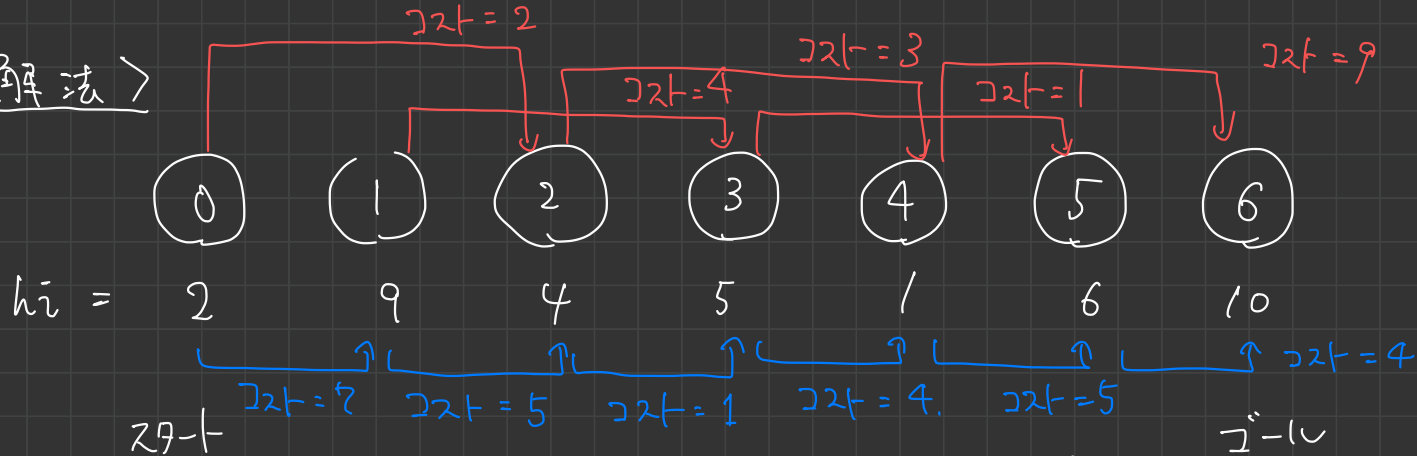
N の足場がある。 i ($0 \leq i \leq N-1$) 番目の足場、高さは h_i である。
初めカエルは 0 番目の足場について、以下のいずれか、行動を繰り返して $N-1$ 番目の足場を目指す。

- 足場 $i \rightarrow$ 足場 $i+1$ へと移動 (コスト = $|h_i - h_{i+1}|$)
- 足場 $i \rightarrow$ 足場 $i+2$ へと移動 (コスト = $|h_i - h_{i+2}|$)

カエルが $N-1$ 番目の足場に辿りつくまでに要するコストの 総和 の最小値を求めよ。

(例) $N=7$. $h_i = \{2, 9, 4, 5, 1, 6, 10\}$

< 解法 >



求めるのはコストの 総和.

$i = 0$ にく → 初めにいる i のコスト = 0.

$i = 1$, → コスト = 7 しかない

$i = 2$, → $\begin{cases} 0 \rightarrow 2 : \text{コスト} = 2 : \text{総和} = 0 + 2 = 2 \\ 1 \rightarrow 2 : \text{コスト} = 5 : \text{総和} = 0 + 5 = 5 \end{cases}$ $7 + 5 = 12$

$$\begin{aligned}
 i=3 & \quad \rightarrow \begin{cases} 1 \rightarrow 3 : \text{Jst} = 4 : \text{Sum} = 7 + 4 = 11 \\ 2 \rightarrow 3 : \text{Jst} = 1 : \text{Sum} = 2 + 1 = 3 \quad \circ \end{cases} \\
 i=4 & \quad \rightarrow \begin{cases} 2 \rightarrow 4 : \text{Jst} = 3 : \text{Sum} = 2 + 3 = 5 \quad \circ \\ 3 \rightarrow 4 : \text{Jst} = 4 : \text{Sum} = 3 + 4 = 7 \end{cases} \\
 i=5 & \quad \rightarrow \begin{cases} 3 \rightarrow 5 : \text{Jst} = 1 : \text{Sum} = 3 + 1 = 4 \quad \circ \\ 4 \rightarrow 5 : \text{Jst} = 5 : \text{Sum} = 5 + 5 = 10 \end{cases} \\
 i=6 & \quad \rightarrow \begin{cases} 4 \rightarrow 6 : \text{Jst} = 9 : \text{Sum} = 5 + 9 = 14 \\ 5 \rightarrow 6 : \text{Jst} = 4 : \text{Sum} = 4 + 4 = 8 \quad \circ \end{cases}
 \end{aligned}$$

2.2. i 番目に辿りつくための総和の最小値は
 (Jst)

$$\begin{array}{ll}
 i=0 & \rightarrow 0 \\
 i=1 & \rightarrow 7 \\
 i=2 & \rightarrow 2 \\
 i=3 & \rightarrow 3 \\
 i=4 & \rightarrow 5
 \end{array}
 \qquad
 \begin{array}{ll}
 i=5 & \rightarrow 4 \\
 i=6 & \rightarrow 8
 \end{array}$$

4