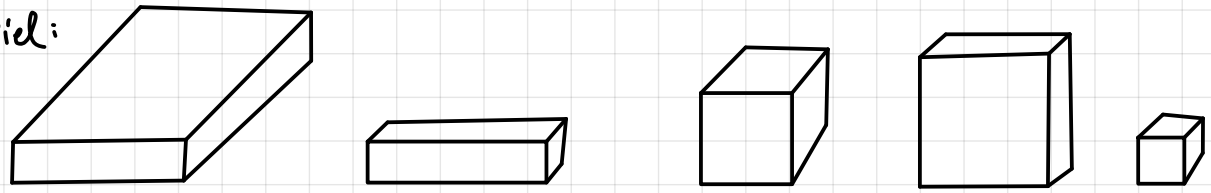
Beispiel:



$l=6$	$l=1$	$l=2$	$l=1$	$l=1$
$b=4$	$b=4$	$b=2$	$b=3$	$b=1$
$h=1$	$h=1$	$h=2$	$h=3$	$h=1$

Wir konstruieren folgenden Algorithmus:

Dimension der DP-Table: $1 \times n$ ($DP[1, \dots, n]$)

Bedeutung der Entries: In entry $DP[i]$ speichern wir die Höhe des höchsten Turms, der den Stein i als oberstes Element ("Turmspitze") hat, ab.

Berechnung eines Entries: Da die Legosteine absteigend (nach Grundfläche) sortiert gegeben sind, kann der Stein i – wenn überhaupt – nur auf steine in $[1, \dots, i-1]$ gesetzt werden. Also gucken wir uns alle möglichen Steine, auf die i gesetzt werden könnte, an, prüfen, ob i auch wirklich passt und wählen den höchsten daraus resultierenden Turm. Falls Stein i auf keinen anderen Stein gesetzt werden kann, wird $DP[i] = h[i]$, also ein Turm, der nur aus Stein i besteht. Formal also:

$$DP[i] = \max_{1 \le k \le i} \{\delta_{ki} \cdot (DP[k] + h[i])\}$$

wobei $\delta_{ki} = \begin{cases} 1, & \text{falls } i \text{ auf } k \text{ passt} \\ 0, & \text{falls nicht} \end{cases}$

Anfangs setzen wir $DP[k] = 0 \quad \forall k \in \{1, \dots, n\}$

Berechnungsreihenfolge: Wir iterieren von links nach rechts über das gegebene Lego-Stein-Array und befüllen das DP array ebenfalls von links nach rechts mit Werten: Denn, um den

höchsten Turm, der den i-ten Stein als Spitze hat, müssen wir die Höhen aller Türme, auf die der i-te Stein gesetzt werden kann, bereits kennen.

Lösung extrahieren: Am Ende steht in jedem Entry DP[i] die Höhe des höchsten Turmes, der Stein i als Spitze hat. Also müssen wir noch einmal durch das Array iterieren und uns den höchsten Wert merken: $maxTurm = \max_{1 \leq i \leq n} \{DP[i]\}$

Laufzeit: Der Algorithmus iteriert einmal über das Array der Legosteine. Für jeden Stein gucken wir uns dann noch einmal die Höhen aller "grösseren" Steine an, auf die dieser theoretisch gesetzt werden könnte.

Also. $T(n) = \sum_{i=1}^{n} \sum_{j=1}^{i} c = \sum_{i=1}^{n} i \cdot c = c \cdot \left(\frac{n(n+1)}{2}\right)$

$\leq O(n^2)$

Abschliessend iterieren wir noch einmal über das DP-Array, um das Maximum zu finden: $O(n)$

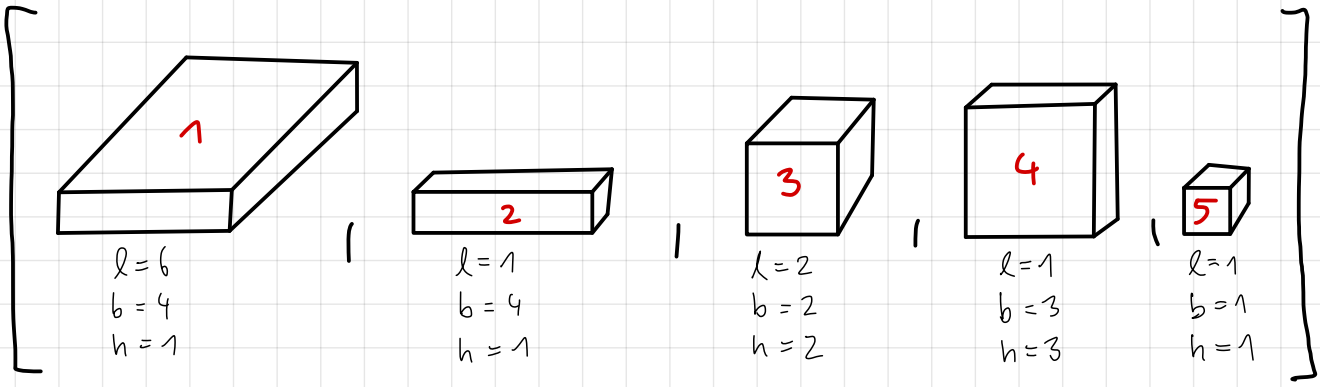
Also insgesamt, da $O(n) \subset O(n^2)$:

$O(n^2)$

Nun zeige ich noch einmal anhand des Beispiels, wie der Algo funktioniert (nächste Seite):

Beispiel:

Gegeben:



Unser DP-Table: $[0, 0, 0, 0, 0]$ (1 2 3 4 5)

1. Iteration:

Wir betrachten nur Stein 1. Da er ganz links steht, gibt es keine Türme, auf die er gestellt werden könnte. Also:

← Höhe des max. Turms mit Stein 1 als Spitze

Unser DP-Table: $[1, 0, 0, 0, 0]$ (1 2 3 4 5)

2. Iteration:

Wir betrachten nun Stein 2. Die einzigen Türme die mit diesem Stein als Spitze möglich sind, sind Stein 2 auf Stein 1 () oder nur Stein 2. Die Höhe der ersten Möglichkeit ist DP[1] + h(2) = 2 > 1 (Höhe der zweiten Möglichkeit). Also:

Unser DP-Table: $[1, 2, 0, 0, 0]$ (1 2 3 4 5)

2. Iteration:

Wir betrachten nun Stein 3. Alle Türme mit Stein 3 als Spitze bauen entweder nur auf Stein 3 oder Türmen mit "größeren" Spitzen, auf die Stein 3 gelegt werden kann. Der Turm mit Stein 2 ist zwar höher als der mit Stein 1 als Spitze, aber da passt Stein 3 nicht

drauf. Aber auf den Turm mit Stein 1 als Spitze. Also:

Unser DP-Table: $[\underset{1}{1}, \underset{2}{2}, \underset{3}{3}, \underset{4}{0}, \underset{5}{0}]$

4. Iteration:

Wir betrachten nur Stein 4. Alle möglichen Türme, auf die dieser Stein gesetzt werden könnte, müssen Spitzen haben die grösser sind und passen. Also gucken wir uns alle Türme mit "grösseren" Spitzen an, also alle links von Stein 4. In Frage kommen also die Türme mit Stein 1, 2, 3 als Spitze. Aber: Auf den Turm mit Stein 3 als Spitze passt Stein 3 nicht. Also nur Spitzen 1, 2. Setzen wir ihn auf Spitze 1, haben wir neuen Turm mit Stein 4 als Spitze und Höhe: $DP[1] + h[4] = 1 + 3 = 4$. Setzen wir ihn auf Spitze 2, haben wir: $DP[2] + h[4] = 2 + 3 = 5 > 4$. Also ist 5 die Höhe des höchsten Turmes mit Stein 4 als Spitze:

Unser DP-Table: $[\underset{1}{1}, \underset{2}{2}, \underset{3}{3}, \underset{4}{5}, \underset{5}{0}]$

Im letzten Schritt füllen wir noch den letzten Eintrag:

Unser DP-Table: $[\underset{1}{1}, \underset{2}{2}, \underset{3}{3}, \underset{4}{5}, \underset{5}{6}]$

Nun iterieren wir noch einmal über das Array und sehen: der höchste Turm hat Höhe 6 und Stein 5 als Spitze. Es ist folgender Turm: