1 Problem 6.2

Let $b[i]$ be the minimum total penalty for stopping at hotel a_i, $1 \leq i \leq n$. We have

$$b[i] = \min_{1 \leq j < i} \{b[j] + (200 - (b[i] - b[j]))^2\}.$$

Also record the value j which yields $\min_{1 \leq j < i} \{b[j] + (200 - (b[i] - b[j]))^2$ and set `b[i].prev = b[j]`. Backtrack from $b[n]$ to get the sequence of hotels to stop by.

2 Problem 6.3

Let $S[i]$ be the maximum total profit we get from building some restaurants in $\{m_1, m_2, \ldots, m_i\}$. Consider 2 cases:

(1) If restaurant m_i should not be built, then $S[i] = S[i-1]$.

(2) If restaurant m_i should be built, then let c_i be the maximum index j which yields $m_i - m_j \geq k$. We then have $S[i] = p_i + S[c_i]$.

Therefore in general,

$$S_i = \max\{S[i-1], p_i + S[c_i]\}.$$

To get the sequence of restaurants, keep an array $R[n]$ such that $R[i] = 1$ if restaurant i is built in the optimal solution, and $R[i] = 0$ otherwise. When we calculate $S[i]$, if the max falls to case (1), $R[i] = 0$. Else, $R[i] = 1$. Output all the $R[i]$s that are 1.

3 Problem 6.4

Consider an array $S[n]$ where $S[i] =$ `true` if the substring $s_1 s_2 \ldots s_i$ is a valid string, and `false` otherwise. We have $S[1] =$ `dict(s[1])` and

$$\begin{aligned} S[i] = (&S[1] \mathbin{\&\&} \texttt{dict(s[2 .. i])} \\ &||(S[2] \mathbin{\&\&} \texttt{dict(s[3..i])} \\ &|| \ldots \\ &||(S[i-1] \mathbin{\&\&} \texttt{dict(s[i..i])})), \end{aligned}$$

where $s[j..i]$ is $s_j s_{j+1} \ldots s_i$.

4 Problem 6.6

Let the input string be $x_1 x_2 \ldots x_n$.
Let $Z = \{a, b, c\}$ and let $T[i, j] \subset Z$ be the set of the possible values that the product $x_i x_{i+1} \ldots x_j$ can yield with all possible parenthesizations.
We see that $T[i, i] = x_i$ for all $1 \leq i \leq n$. We need to compute $T[1, n]$.
Define $A \times B$ as $\{a \cdot b | a \in A, b \in B\}$.
We note that $T[i, i+1] = T[i, i] \bigcup T[i+1, i+1]$ and $T[i, i+2] = (T[i, i] \times T[i+1, i+2]) \bigcup (T[i, i+1] \times T[i+2, i+2])$ (to put it another way, abc can be written as $(a)(bc)$ or $(ab)(c)$).
We therefore see that we already have

$$T[1, 1], T[2, 2], T[3, 3], \ldots,$$

from which we can calculate

$$T[1, 2], T[2, 3], T[3, 4], \ldots,$$