Problem 3. [17 points] **Number Scrabble**

Number Scrabble is a one-player game played on an array $T = [t_0, \ldots, t_{n-1}]$ of n positive integers. There is a list $P = \{(p_0, v(p_0)), \ldots, (p_{m-1}, v(p_{m-1}))\}$ of m unique **playable words**, where playable word p_i is a non-empty array of at most 10 positive integers and $v(p_i)$ is the positive integer value of p_i. The objective of the game is to find a **gameplay** S — a list of **non-overlapping** subarrays (i.e., substrings) of T, each a playable word — where S has maximum total value, $\sum_{s \in S} v(s)$. For example, if

$$T = [1, 5, 2, 4, 1] \text{ and } P = \{([2], 3), ([1], 1), ([5, 2], 8), ([1, 2], 12), ([1, 5], 2)\},$$

then $S_1 = ([1, 5], [2], [1])$, $S_2 = ([1], [5, 2], [1])$, and $S_3 = ([1], [2], [1])$ are all valid gameplays, with total values 6, 10, and 5 respectively. Note playable word $[1, 2]$ cannot exist in any gameplay of T, since $[1, 2]$ is not a contiguous subarray of T. Given T and P, describe an $O(n + m)$-time algorithm to return a gameplay of maximum total value.

已知一 $P = \{(p_0, v(p_0)), \ldots, (p_{m-1}, v(p_{m-1}))\}$ 和一 $T = [t_0, t_1, \ldots, t_{n-1}]$

欲找到一 $S = (S_1, S_2, S_3 \ldots, S_k) \rightarrow \sum_{s \in S} v(s)$ 最大

① optimal substructure?

定義子問題為: $d[i]$ 為 $T[i:]$ 下之 optimal value

Recursive relation:

$$d[i] = \begin{cases} 0 & \text{if } i = n \\ \max\{d[i+1], \max\limits_{\substack{0 \le j \le 10 \\ i+j \le n}} \{d[i+j+1] + v(T[i:i+j])\}\} & \text{otherwise} \end{cases}$$

$T[i:i+j] \in P$ ^ 需查找 $T[i:i+j] \in P$, 可用 bloom filter 或 hash table

Example: $T = [1, 5, 2, 4, 1]$

p_i	[2]	[1]	[5, 2]	[1, 2]	[1, 5]
$v(p_i)$	3	1	8	12	2

i	5	4	3	2	1	0
d	0	1	1	4	9	10

Time Complexity: $O(n) \times O(1) = O(n)$

$\because$ playable word 的 size 至多為 10

$\Rightarrow O(n+m)$ 其中 $O(m)$ 為建構 hash table 花的時間