12. (15 points) There are n stations along a coastal railway ($n > 0$). You're planning to select some of them to open cafes. Three arrays S, L and R have been given, including

- $S = (s_1, s_2, ..., s_n)$: the list of the stations from s_1 (first) to s_n (last).
- $L = (l_1, l_2, ..., l_n)$: the locations of the stations, where l_i is the distance of s_i from the first station s_1. So $l_1 = 0$ and l_n is the length of the railway. $l_1 < l_2 < ... < l_n$.
- $R = (r_1, r_2, ..., r_n)$: the revenues of the cafes, where r_i (> 0) is the revenue for opening a cafe in s_i.

The only one constraint in your plan is that the distance of any pair of your selected stations should be longer than a given threshold T. If s_i and s_j ($i \neq j$) are selected, the total revenue would be $l_i + l_j$. Different selection leads to different total revenue. Given S, L, R and T, your goal is to pick up a subset of the stations to maximize the total revenue under the constraint. Suppose that $f(n)$ returns the maximum total revenue for the cafes you select from the first n stations. Please answer the following questions. **No code is required (code will not be graded).**

(a) (9 points) Give an $O(n^2)$ solution by defining a recurrence formula for $f(n)$. Clearly explain the meaning of your formula and why it can be computed in $O(n^2)$ time.

(b) (6 points) Consider the special case that the distance difference between two consecutive stations is 1, i.e., $l_{i+1} - l_i = 1$ ($1 \leq i \leq n-1$). Give an $O(n)$ solution by defining a recurrence formula for $f(n)$. Clearly explain the meaning of your formula and why it can be computed in $O(n)$ time.

(a). 定義子問題為 d[i], 在 s[1,...,i] 下包含 s[i] 可得之 max revenue

Example:

i	1	2	3	4	5
l_i	0	2	4	5	7
r_i	2	3	1	4	2

且 T=2

可知 Optimal value 為: 11

$$d[i] = \begin{cases} r_0 & \text{if } i = 0 \\ \max\limits_{\substack{0 \le k \le i-1 \\ l[i]-l[k] \ge T}} \{ d[k] + r[k] \} & \end{cases}$$

i	1	2	3	4	5
d_i	2	5	6	9	11

(b). 定義子問題為 d[i], 在 s[1,...,i] 下包含 s[i] 可得之 max revenue

$$d[i] = \begin{cases} 0 & \text{if } i = 0 \\ r_1 & \text{if } i = 1 \\ d[i-T] + r[i] & \text{otherwise} \end{cases}$$

Example:

i	1	2	3	4	5
l_i	0	1	2	3	4
r_i	1	3	1	4	2

且 T=2

i	1	2	3	4	5
d_i	1	3	2	7	4

max{ d[1], ..., d[n] } = 7 為所求, 為 O(n)