(A)

Idea

Recall that $C(i)$ means the minimum penalty for sequence $\langle a_1, .., a_i \rangle$. Since each subsequence could now be at most 10 elements, the previous subproblem we store and calculate could be limited to at most 10 for each subproblem. This is a constant factor and could reduce both complexity by an order of $O(n)$

Subproblem:

$C(i)$ is the minimum penalty for line-breaking the sequence $\langle a_1, .., a_i \rangle$, providing the additional constraint given by (a). The final answer is $C(n)$ obviously.

Recursive Formula:

Notice that we need to precompute a prefix sum $P(i) = \sum_{k=1}^{i} a_i$, from 1 to n. This will take only $O(n)$ and later on the sum $S(i,j) = P(i) - P(j)$. But we do not need to keep all term $P(i)$ available. As $C(i)$ depends at most 10 terms before, we could compute $P(i)$ as we come to $C(i)$ and discards the prefix sum that goes beyond 10 terms. Since the last subsequence has at most 5 elements, we need to treat it differently. Therefore

$$C(0) = 0 \qquad C(i) = \begin{cases} \min_{j \in [\max(0, i-10), i-1],\ S(i,j) \leq L} (C(j) + f(L - S(i,j))) & \text{if } i < n \\ \min_{j \in [\max(0, i-5), i-1],\ S(i,j) \leq L} (C(j) + f(L - S(i,j))) & \text{if } i = n \end{cases}$$

j in here act as the break index, therefore $j \in [\max(0, i-10), i-1]$ effectively ensures that new subsequence $\langle a_{j+1}, .., a_i \rangle$ have length at most 10 elements.

Evaluation Order

The evaluation order should be i from 0 to n obviously.

Complexity

Since we compute all $C(i)$, we have $O(n)$ tasks. For each task, the evaluation takes at most 10 iterations, for each iterations, the computation takes at most $O(1)$ time, as f are constant time. So, each task takes at most $O(1)$ and therefore all tasks takes $O(n)$. Adding the time for prefix sum $O(n)$, the time complexity is $O(n)$. The space complexity should be $O(1)$, as we only need to keep at most 10 terms prefix sum and previous results (which are constants).

(B)

Idea

We will introduce a variable l to track the number of subsequence used.

Subproblem

$C(i, j)$ is the minimum penalty for line-breaking the sequence $\langle a_1, .., a_i \rangle$ using exactly j subsequence. The final answer is $C(n, m)$ obviously.

Recursive Formula:

For base case, we want $C(0, 0) = 0$, as zero length, zero subsequence sequence have 0 penalty. Also, we want $C(i, 0) = \infty$ which $i > 0$, as it's impossible to have non-zero length sequence but have zero subsequence. $C(i, j) = \infty$ which $i < j$, as it's impossible to have more subsequences than total length of the sequence (i.e., we don't want zero length subsequence).

Fro recursive case where $i \geq j \geq 1$

$$C(i, j) = \min_{k \in [0, i], S(i,k) \leq L} (C(k, j-1) + f(L - S(i, k)))$$

Evaluation Order

First we compute the prefix sums. Then iterate i, j in increasing order, as later i, j depends on smaller i, j cases.

Complexity

We have total of $O(mn)$ tasks as we iterate i, j. For each tasks, we iterate through k so it takes $O(n)$ time as f is constant. So we spend total of $O(m \cdot n^2)$ and additional $O(n)$ for prefix sum. Total time complexity is $O(m \cdot n^2)$. Since we store $C(i, j)$ so it takes $O(m \cdot n)$ obviously.