Сначала заменим этот массив массивом префикс-сумм. То есть заполним массив по правилу: $prefix[i] = \sum\limits_{j=1}^{i} a[j] = \begin{cases} a[i] & i = 0 \\ prefix[i-1] + a[i] & i \neq 0 \end{cases}$. Это займет $\mathcal{O}(n)$ времени, так как мы делаем один проход по массиву.

Теперь заполним массив a (который сейчас выполняет роль префикс-сумм) новыми значениями. Вычислять будем с конца. Допустим сейчас мы обрабатываем i элемент, тогда: вычисляем $f(i)$ и берем сумму на отрезке при помощи префикс-сумм ($\sum\limits_{i=l}^{r} a[i] = \begin{cases} prefix[r] & l = 0 \\ prefix[r] - prefix[l-1] & l \neq 0 \end{cases}$, где l и r, $f(i)$ и i соответственно). Докажем корректность, а именно что при подсчете i элемента префикс-суммы будут корректны. Это так потому что $1 \leq f(i) \leq i$ по условию, а в решении мы подсчитываем с конца, т.е. идем от больших индексов к меньшим, следовательно все элементы массива с индексами меньшими или равными i будут неизменными, то есть там будут корректные значения. Вычисление $a'[k]$ занимает $\mathcal{O}(1)$ времени и происходит это n раз, следовательно итоговая асимптотика $\mathcal{O}(n + n) = \mathcal{O}(2n) = \mathcal{O}(n)$ времени.
Краткий алгоритм решения можно записать так:

1. Заменяем массив a массивом префикс-сумм.
2. Вычисляем значение $a'[i]$ (i от больших к меньшим):
   (a) Вычисляем $f(i)$.
   (b) Вычисляем сумму на отрезке $[f(i); i]$ при помощи префикс-сумм.
3. Заменяем $a[i]$ вычисленным значением $a'[i]$ (которое мы хранили в отдельной переменной).

Из алгоритма понятно, что используется $\mathcal{O}(1)$ доп. памяти.