1 Largest Ad Gap

Problem: Let A be a sorted (ascending) list of cue points with size $|A| = m$, and r be the number of cue points we aim to remove from this list A. We define a gap between any pair of consecutive items A_i, A_{i+1} $(1 \leq i < m)$ as the difference $A_{i+1} - A_i$.

Let $O_{m,r}$ be a sorted list of cue points $|O_{m,r}| = m - r$, $O_{m,r} \subset A$ such that the smallest gap in $O_{m,r}$ is maximized.

Write an algorithm that returns the value $D(m, r)$, the minimum gap between any pair of consecutive numbers in $O_{m,r}$. That is, write an algorithm which selects the optimal r elements to remove from A in order to the maximize the smallest gap in the resulting list $O_{m,r}$.

Note: Notice that here we aim to return the minimum gap between each consecutive pair in $O_{m,r}$, and not the set $O_{m,r}$ itself. We assert that given a dynamic programming table D (as described below), the polynomial time method of tracing can be employed to find $O_{m,r}$. Other methods (e.g. storing the optimal sub-list $O_{n,k}$ at each step) may also be used.

We aim to solve this problem by method of dynamic programming table (i.e. a memoization, or caching solution). Let D be a dynamic programming table of size $m \times r$. We aim to fill the table D such that each value of the table, $D(n, k)$, is an optimal solution for $0 \leq n \leq m$ and $0 \leq k \leq r$. In other words, each $D(n, k)$ is largest minimum gap formed by removing k elements from the list $A_1 \ldots A_n$.

First, let us consider the various trivial solutions for specific pairs n, k.

1. Consider the case where, after removing k elements from A there is only a single element remaining in the resulting list, $|A'| = 1$. If there is only a single element in the resulting list, we consider the list trivially optimal, and denote it with ∞. That is,
$$\forall n, k \text{ such that } n - k \leq 1, \quad D(n, k) = \infty \tag{1}$$
This fact allows us to fill in the cells of our dynamic programming table D which fall on and above its diagonal.

2. Next, consider the trivial case where $k = 0$. That is, the case where there are no elements to remove from A. When $k = 0$ and $n = 2$, we define the maximum gap between consecutive elements in A, as
$$D(2, 0) = A_2 - A_1 \tag{2}$$
From here we can build on the preceding equation by noting that
$$\forall n > 2 \text{ such that } k = 0, \quad D(n, 0) = \min(A_n - A_{n-1}, D(n-1, 0)) \tag{3}$$
This new fact, coupled with equation (2) allows us to inductively fill in the leftmost column of D, which is of the form $D(n, 0)$.

Now we will derive an equation to find the value of $D(n, k)$ for all remaining n, k.

Recall that $D(n, k)$ is the optimal solution of the subproblem which considers only the first n items of A and removes only k elements from that subarray (where $n \leq m$, $k \leq r$).

First, we assert that the *nth* element of A is always present in the optimal solution list $O_{n,k}$ (that is, A_n will *not* be removed). To show this, consider the following two cases, one of which must be true for the new element A_n:

1. A_n and the preceding element A_{n-1} form the smallest gap in $O_{n,k}$ (i.e. $A_n - A_{n-1} = D(n, k)$). In this case, this minimum gap may be removed by eliminating the element A_n, but a gap of larger or equal value may *always* be produced by removing the preceding element A_{n-1}. This is because, since A is sorted, $A_n - A_{n-2} \geq A_n - A_{n-1}$.

2. A_n and the preceding element A_{n-1} **do not** form the smallest gap in A. If this is the case, then A_n is trivially in $O_{n,k}$, since it would be better to remove an element in the smallest gap of A.

Next, we form the following injunction:

For any list of size n the largest gap of $A_1 \ldots A_n$ after removing k elements, $D(n, k)$, must be the largest of:

- $\min(A_n - A_{n-1}, D(n-1, k))$
- $\min(A_n - A_{n-2}, D(n-2, k-1))$
- $\min(A_n - A_{n-3}, D(n-3, k-2))$

- ...
- $\min(A_n - A_{n-(k+1)}, D(n-(k+1), k-k))$

Using mathematical notation, we may re-write the above as,

$$D(n,k) = \max_{0 \le i \le k} (\min(A_n - A_{n-(i+1)}, D(n-(i+1), k-i))) \tag{4}$$

Explanation: Consider the list $A_1 \dots A_{n-1}$, which has an optimal solution $D(n-1,k)$ formed by calculating the minimum gap in $O_{n-1,k}$. Now observe the following: upon adding the newly revealed value A_n to the end of the list $O_{n-1,k}$, one of the two following cases **must** be satisfied. Either,

- The new item A_n forms the new minimum gap of the list $A_1 \dots A_n$. That is, the minimum gap is $A_n - A_{n-(i+1)})$ for some i such that $0 \le i \le k$.
- The new item A_n **does not** form the new minimum gap of $A_1 \dots A_n$. That is, the minimum gap of $A_1 \dots A_n$ is $D(n-(i+1), k-i)$ for some i such that $0 \le i \le k$.

So, to find the largest minimum gap, we take the largest of the above cases for all valid i, as denoted in (4).

The runtime of the above algorithm is $O(m \cdot r^2)$, since iterating through the table D takes $O(m \cdot r)$ time, and each iterative step takes $O(r)$ time.