Solution for Game

This is quite obviously a task where dynamic programming shall be used – there are exponentially many ways the game can be played, but only polynomially many states in which the game can be. More precisely, at any moment the current state of the game is a contiguous subsequence of the original sequence of cities. In other words, each state can be uniquely described by the smallest index A of a city you have, and the smallest index B>A of a city you don't have. We will denote the state represented by A and B as [A,B).

Your goal at any moment is the same: maximize your profit in the game played from the current state. Let $S_{A,B}$ be the sum of profits for cities in [A,B) and $P_{A,B}$ be the best profit you can get from the game played on [A,B). In our solution, we will compute all the values $P_{A,B}$, and for each of them the optimal first move. Using this information we can then play the game optimally.
Note that our opponent can compute the same values and use them for his optimal play as well – when presented with a choice, he will always pick the option that will give us a lower profit. We can use this observation to write a recurrence relation for the values $P_{A,B}$:
$P_{A,A} = P_{A,A+1} = 0$
$P_{A,B} = \max_C(\min(S_{A,C}+P_{A,C}, S_{C,B}+P_{C,B}))$
The second equality holds for B>A+1, where the maximum is taken over all C such that A<=C<=B.
An explanation in words: When we pick the place C where to place the bastion, the opponent will examine the two possibilities and leave us with the worse one. Knowing this, we can compute the profit for each possible choice of C, and pick the best one.

By using the above recurrence, one can easily compute a single value $P_{A,B}$ in O(N), and as there are $O(N^2)$ pairs A,B, the total time complexity of the precomputation is $O(N^3)$. We can then make moves in O(1) each.

We can easily precompute the values $S_{A,B}$ in time $O(N^2)$. However, note that the values $S_{A,B}$ can be precomputed in O(N) time with O(N) memory – it is enough to store the values $S_{0,X}$, as $S_{A,B}=S_{0,B}-S_{0,A}$.

To get a better time complexity, we need to find a way to limit the set of places C we need to examine when computing $P_{A,B}$. We will use the following observation:
(*) Suppose that for the segment A,B the optimal split is C. Then for the segment A,B+1 the optimal split is at least C.
We will prove this later.

Once we trust that (*) holds, we can compute the values $P_{A,B}$ and $C_{A,B}$ in the following order:
[0,1) [1,2) ... [N-2,N-1) [N-1,N)
[0,2) [1,3) ... [N-2,N)
...
[0,N-1) [1,N)
[0,N)
When computing $P_{A,B}$ and $C_{A,B}$, we know from (*) and from symmetry that it is enough to consider $C_{A,B}$ between $C_{A,B-1}$ and $C_{A+1,B}$, inclusive.
For each row of our table above, the total number of possibilities we have to try when computing it in this way is at most 2N. If this is not clear to you, consider the following example:
The best split for [0,7) is between the best split for [0,6) and the best split for **[1,7)**.
The best split for [1,8) is between the best split for **[1,7)** and the best split for [2,8).
The best split for [2,9) is between the best split for [2,8) and ...

Thus we get a solution with both time and memory complexity $O(N^2)$.

Lemma 1:
For all A, B we have $P_{A,B+1} >= P_{A,B}$.
This is obvious. A formal proof by induction is possible, based on the fact that making the same split for [A,B+1) as for [A,B) leads at least to the same profit.

Proof of the observation (*):
Let C be an optimal split for [A,B). We will show that for [A,B+1) there is some optimal split >= C.
Consider the values $X = S_{A,C}+P_{A,C}$ and $Y = S_{C,B}+P_{C,B}$.
We have $P_{A,B} = \min(X,Y)$, and from Lemma 1 we have $P_{A,B+1} >= \min(X,Y)$.

Case 1: $X <= Y$.
In this case $P_{A,B} = X$. Let $D < C$. Clearly $S_{A,D} < S_{A,C}$. From Lemma 1, $P_{A,D} <= P_{A,C}$. Therefore $S_{A,D}+P_{A,D} < S_{A,C}+P_{A,C}$. If we make a split at D, our opponent will leave us with the part [A,D), and thus our total profit will be less than X. Thus such D can not be an optimal split, as $P_{A,B+1}$ must be at least X.
Case 2: $X > Y$.
In this case $P_{A,B} = Y$. Let $D < C$. The split at C was optimal for [A,B), therefore the split at D was equal or worse. This means that $\min(S_{A,D}+P_{A,D}, S_{D,B}+P_{D,B}) <= Y$. Clearly, from $D<C$ it follows that $S_{D,B}+P_{D,B} > S_{C,B}+P_{C,B}=Y$ for the same reasons as in the previous case. Thus we must have $S_{A,D}+P_{A,D} <= Y$.
This means that if we split [A,B+1) at D, our opponent can leave us with [A,D), which will give us total profit at most Y. However, the split at C will give us total profit more than Y, therefore again D can not be an optimal split.