三段跳び (解説)

ある 2 つの index i, j が,（いずれかのクエリにおいて）$a = i,\ b = j$ として使われる可能性があるかどうかについて考える．

まず，k であって，$i < k < j,\ A_k \geqq A_j$ を満たすものが存在する場合，$a = i,\ b = j$ となる解を考慮する必要はない．(なぜなら，$b = k$ とすれば損しない解が得られるから．)

次に，$A_i \leqq A_j$ であり，かつ，ある k が存在して $i < k < j,\ A_i \leqq A_k \leqq A_j$ を満たす場合を考える．このとき，$a = i,\ b = j$ となる解を考慮する必要はない．(なぜなら，$a = k$ とすれば損しない解が得られるから．)

以上 2 つのことを考慮すると，考えるべき i, j のペアは $O(N)$ 個になっている．これらのペアは，i を N から 1 まで変化させながら，候補となる j の集合を保持すれば列挙できる．これで，a, b の候補となる，i, j のペアが列挙できる．

あとはクエリの処理を行う．添字 t を N から 1 まで変化させながら，数列 v を保持することにする．数列 v は，その x 番目の値が，$t \leqq a < b < c = x$ となる a, b, c $(b - a \leqq c - b)$ についての $A_a + A_b + A_c$ の最大値となるようにする．

この数列 v が保持できればクエリに答えるのは簡単である．

t を $s + 1$ から s へ変化させたときに数列 v がどう変化するかを考える．これは，先程列挙した a, b の候補のペア i, j であって $s = i$ となるものについて，$2j - i \leqq k$ となる全ての k について，

$$v_k = \max(v_k, A_i + A_j + A_k)$$

とできればよい．

Segment Tree を用いると 1 つの候補についての更新が $O(\log N)$ でできる．また，各クエリについて答えを求めるのも 1 クエリあたり $O(\log N)$ でできる．よって，この問題は $O((N + Q)\log N)$ で解ける．