Cruise: solution

The first observation is that the optimal cruise will always be the convex hull of a subset of the islands (including Piraeus).

Now, the objective function $\frac{P}{D}$ is hard to deal with directly, since it is not linear (Here P is the sum of points collected and D the total distance). So, instead of directly finding the optimal ratio $\frac{P}{D}$, we can ask the question $\frac{P}{D} \geq C$?, for some C. This is equivalent to checking $P - C \cdot D \geq 0$, which is linear! So it suffices to find the maximum value of $f = P - C \cdot D$, over all possible routes, since then we can find the answer by binary searching for the maximum C.

This is much better, since we can use dynamic programming to find the maximum value. Sort all islands by angle around the origin (Piraeus). (Suppose 1 is the index of the origin). Let $dp(i)$ be the maximum value of f for routes that begin at 1, end at i, and use only islands in $\{1, \ldots, i\}$. Then we have that $dp(i) = max_{j<i}(dp(j) + cnt(i,j) - C \cdot dist(i,j))$, where $cnt(i,j)$ is the total sum of points inside the triangle $(1,i,j)$ and $dist(i,j)$ the euclidean distance between islands i and j. Directly applying this recurrence relation, we get a solution with complexity $O(N^3 \log ANS)$ that scores 58 points.

In order to improve it further, we have to precompute the values $cnt(i,j)$. It turns out that we can do this in time $O(N^2 \log N)$. We just have to sum the points of islands that lie inside the triangle $(1,i,j)$. In fact, we need to sum islands that are between i and j in our angle ordering, and also lie "to the left" of the segment (i,j). The first two can be ensured by the order in which we compute the values $cnt(i,j)$. The third one needs some data structure that can maintain sums. The simplest option is to use a Binary Indexed Tree, indexed by angle. In order to use it, for fixed i, we should first sort points by angle around i, and then just query the BIT for the sum of points k, such that $angle(i,k) \leq angle(i,j)$, and update accordingly.

This solution has complexity $O(N^2 \log ANS)$ and scores 100 points.