High Tide

This is a **regular task**. You must submit a PDF, which can be produced using the LaTeX template on Moodle, exported from a word processor, hand-written or any other method.

This task is **very hard**. It is intended to challenge top students, and should be among your *last* priorities.

You are planning a surfing tour to n different beaches. Your tour will start at time $t = 0$, and end at time $t = 1$. As flights are expensive, you can only visit each beach once, and only one at a time. However, you can visit the beaches in any order, and for any amount of time.

For each of the beaches, you have a chart showing the height of the tide at each point in time. These heights are all continuous functions, and only take non-negative values.

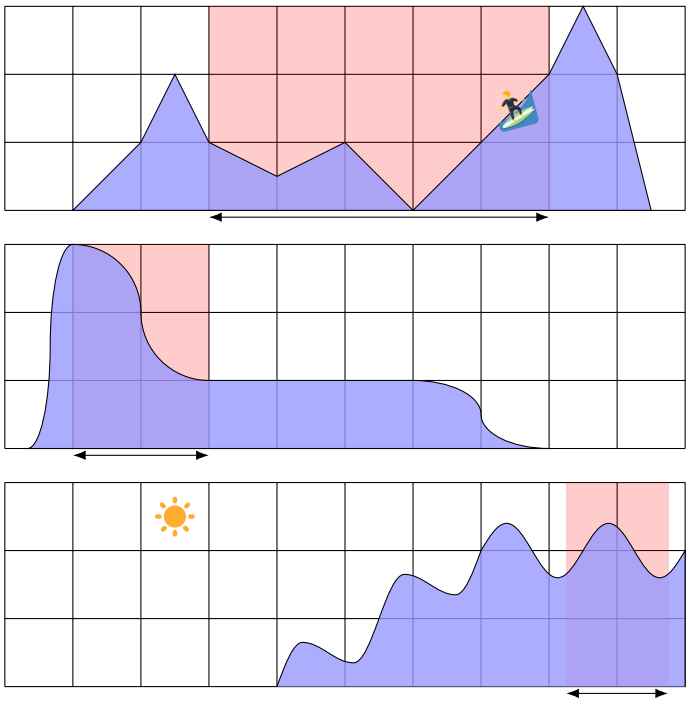
The *tubularity* of a beach B between two times a and b, denoted $B(a, b)$, is the area under the tide chart between those times (in other words, the integral of the tide height from a to b). The tubularity for each beach between $t = 0$ and $t = 1$ is exactly 1, i.e. $B(0, 1) = 1$ for all B.

To ensure that you experience each beach fully, you would like to ensure that the tubularity while you are at each beach is at least $1/n$. Fortunately, you have a very convenient app that lets you choose a beach, an arrival time and a tubularity amount and tells you how long you would need to stay there to experience that amount of tubularity.

Design an algorithm that determines when to arrive and depart at each beach so that you experience a tubularity of at least $1/n$ at each beach, using *no more than n^2 calls to the app*. Assume that you can travel between beaches instantly; that is, you can set the start time for the next beach to be the same as the end time of the current beach.

When proving that your algorithm is correct, it suffices to show that your algorithm will find a solution whenever such a solution exists. This is in contrast to many earlier greedy problems in which you prove that your algorithm gives the *best possible answer*, something you *do not* need to show here.

For example, consider the three tide charts below, showing Manly, Bondi and Coogee beach, respectively. Each tide chart has a total area of 1, so each square represents a tubularity of 1/9. The red areas show a possible schedule, staying at Bondi from $t = 0.1$ to 0.3, then at Manly from $t = 0.3$ to 0.8, then finally at Coogee from $t = 0.825$ to 0.975. We can confirm that this is a valid schedule: $B(0.1, 0.3) = 4/9$, $M(0.3, 0.8) = 4/9$ and $C(0.825, 0.975) = 3/9$, so the tubularity experienced at each location is at least 1/3.



Advice.

- **Correctness**: There are algorithms that correctly solve this problem without being "optimal", i.e. the algorithm finds an order that reaches a tubularity of 1 even though there is another order that reaches the same tubularity earlier. If you try to prove that such an algorithm is **optimal** (using Greedy Stays Ahead or Exchange Argument), your proof will have errors, so you need to use a different method to prove its correctness.
- **Time complexity**: For this problem, we don't require a typical time complexity analysis. Rather the goal is just to argue that the number of calls to the app is at most n^2 as an explicit upper bound, not a big-Oh bound.
- **Expected Length:** Around a page and a half.

Solution.

Attribution.