

# Delivering Pizzas

This is a **regular task**. You must submit a PDF, which can be produced using the L<sup>A</sup>T<sub>E</sub>X template on Moodle, exported from a word processor, hand-written or any other method.

Anzac Parade has  $n$  buildings, each of which has a location in kilometres from the start of the road. You are given an array `Buildings[1..n]`, where `Buildings[i]` is the distance from the start of the road to the  $i$ th building.

The occupants in each building are hungry. Therefore, over each of the next  $n$  days, you will be asked to deliver one pizza to each building between two points  $l$  and  $r$  (inclusive) along Anzac Parade, given in kilometres from the start of the road. This is given as an array `Deliveries[1..n]`, where `Deliveries[i]` contains the pair  $(l, r)$  of points for the  $i$ th day. You would like to design an efficient algorithm to determine exactly how many pizzas you should bring each day.

For example, if `Buildings` = [7, 4, 6, 2, 8] and `Deliveries` = [(1, 3), (5, 5), (1, 9), (8, 8), (2, 6)], then:

- On day 1 you will need 1 pizza, because only the building 2 km from the starting point is between 1 km and 3 km from the starting point,
- You do not need any pizzas on day 2, because there is no building 5 km from the starting point,
- On day 3, you need 5 pizzas, because all 5 buildings are between 1 km and 9 km from the starting point,
- On day 4, you need 1, because only the building 8 km from the starting point is between 8 km and 8 km (inclusive),
- On day 5, you need 3, because the buildings at locations 2 km, 4 km and 6 km from the starting point are between 2 km and 6 km.

Design an algorithm that runs in  $O(n \log n)$  time and determines the number of pizzas you need to bring on each day.

## Advice.

**Expected length:** Up to half a page.