Sections of each loop:
max time for finding current_customer = (numberOfCustomers – 1)(numberOfCustomers)
(it loops in order from index 0 to numberOfCustomers so the maximum length is that it has to traverse the whole thing, i.e. the safe path for n customers is n, n-1, n-2 ..., 1, 0.)

Updating the work array when it is a successful loop
= (numberOfCustomers)(numberOfResources)

once the safe path is complete (finish array is all true) loop numberOfCustomers times to check through.

Therefore time complexity is

O((numberOfCustomers – 1)(numberOfCustomers)
+ (numberOfCustomers)(numberOfResources)
+ numberOfCustomers)

= O((numberOfCustomers – 1)(numberOfCustomers)
+ (numberOfCustomers)(numberOfResources))

therefore it depends on the number of customers vs resources.