```
function get_opt_intervals(intervals, ϵ, y_best)
    stack = Interval[]
    for (x, pq) in intervals
        if !isempty(pq)
            interval = DataStructures.peek(pq)[1]
            y = interval.y

            while length(stack) > 1
                interval1 = stack[end]
                interval2 = stack[end-1]
                x1, y1 = vertex_dist(interval1), interval1.y
                x2, y2 = vertex_dist(interval2), interval2.y
                ℓ = (y2 - y) / (x2 - x)
                if y1 <= ℓ*(x1-x) + y + ϵ
                    break
                end
                # remove previous interval
                pop!(stack)
            end

            if !isempty(stack) && interval.y > stack[end].y + ϵ
                # skip new interval
                continue
            end

            push!(stack, interval) # add new interval
        end
    end
    return stack
end
```

Algorithm 7.11. A routine for obtaining the potentially optimal intervals, where `intervals` is of type `Intervals`, `ϵ` is a tolerance parameter, and `y_best` is the best function evaluation.