```
function get_opt_intervals(intervals, ϵ, y_best)
    stack = Interval[]
    for (d, pq) in intervals
        if !isempty(pq)
            interval = DataStructures.peek(pq)[1]
            x, y = d, interval.y
            while !isempty(stack)
                interval1 = stack[end]
                x1, y1 = vertex_dist(interval1), interval1.y
                L1 = (y - y1)/(x - x1)
                if y1 - L1*x1 > y_best - ϵ || y < y1
                    pop!(stack)
                elseif length(stack) > 1
                    interval2 = stack[end-1]
                    x2, y2 = vertex_dist(interval2), interval2.y
                    L2 = (y1 - y2)/(x1 - x2)
                    if L2 > L1
                        pop!(stack)
                    else
                        break
                    end
                else
                    break
                end
            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.