1. use priority queue to maintain the lowest point on the boundary, and repeatedly perform floodfill starting from the lowest point. $O(sort(nm))$.
2. sorting+union find data structure, merge the regions (with the same height) from low to high. $O(sort(nm)+nm \cdot \alpha(nm))$.
3. reduce to single source bottleneck shortest path (minimax path, minimizing the weight of the maximum-weight edge in the path). The graph is undirected, so we can reduce to minimum spanning tree. expected $O(nm)$ [1].
4. let t be a parameter. merge the regions until each region has size $\geq t$, by finding the minimum weight edge incident to that region using brute force (each region with size $O(t)$ has degree $O(t)$). maintain the regions using union find data structure. then shrink each region into a single node, the new graph is planar, and by Euler's formula we know $|E| = O(|V|) = O(\frac{nm}{t})$. then use algorithm 1 on the new graph in $O(sort(\frac{nm}{t}))$ time. set $t = \sqrt{\frac{sort(nm)}{nm}}$, $O(\sqrt{nm \cdot sort(nm)} + nm \cdot \alpha(nm))$.

References

[1] David R Karger, Philip N Klein, and Robert E Tarjan. A randomized linear-time algorithm to find minimum spanning trees. *Journal of the ACM (JACM)*, 42(2):321–328, 1995.