This is the bottleneck edge problem (or path minimum queries on the MST), which can be preprocessed in $O(n+m)$ time and then answer online queries in $O(1)$ time (for integer weights in word-RAM) [2, 1].
idea: first find the MST in $O(n+m)$ time [3]. Partition the MST into subtrees with size $O(\log n)$. For each of the $O(\frac{n}{\log n})$ roots of the subtrees, precompute the minimum value on the path from it to its $2^i$-th ancestor, for all $O(\log n)$ possible $i$'s, in $O(\frac{n}{\log n} \cdot \log n) = O(n)$ time. Use ladder decomposition and static RMQ on each ladder to support the path minimum query from the $2^i$-th ancestor (for such largest $i$) to the LCA, in $O(n)$ preprocessing time and $O(1)$ query time. For queries within a subtree, use Pătraşcu's result [4] to sort the edge weights within the subtree in linear time during preprocessing, then use word tricks to answer in $O(1)$ time.

# References


[1] Timothy M Chan, Meng He, J Ian Munro, and Gelin Zhou. Succinct indices for path minimum, with applications to path reporting. In *European Symposium on Algorithms*, pages 247–259. Springer, 2014.

[2] Erik D Demaine, Gad M Landau, and Oren Weimann. On cartesian trees and range minimum queries. In *International Colloquium on Automata, Languages, and Programming*, pages 341–353. Springer, 2009.

[3] 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.

[4] Mihai Pătraşcu and Mikkel Thorup. Dynamic integer sets with optimal rank, select, and predecessor search. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 166–175. IEEE, 2014.