sort according to height in decreasing order in $O(sort(n))$, then insert people one by one, according to k. trivially this takes $O(n \log n)$.
we can reduce to the list indexing problem [2] and solve in $O(n \frac{\log n}{\log \log n})$.
in the offline setting we can also reduce to dynamic selection, by inserting people according to height in increasing order, initially insert $1, \dots, n$ in the set, each time query the k-th smallest element in the set and delete it. the running time is $O(n\sqrt{\log n} \log^{\frac{1}{4}} \log n)$ [1].

References

[1] Timothy M Chan and Mihai Pătraşcu. Counting inversions, offline orthogonal range counting, and related problems. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 161–173. Society for Industrial and Applied Mathematics, 2010.

[2] Paul F Dietz. Optimal algorithms for list indexing and subset rank. In *Workshop on Algorithms and Data Structures*, pages 39–46. Springer, 1989.