# FTP_Alg_Week 5: Exercises

jungkyu.canci@hslu.ch

17 October 2023

**Exercise 1** *We apply BUILDKDTREE*$(P, 0)$ *(the pseudocode given in Lecture 8) to the following set* $P$ *of points of the plane:*

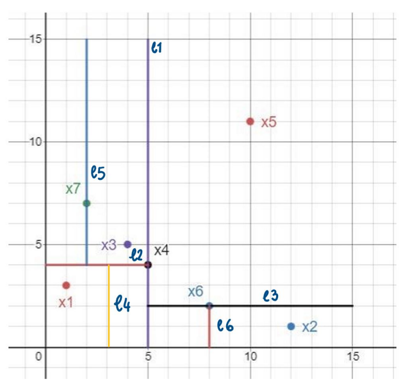
$$P = \{(1,3), (12,1), (4,5), (5,4), (10,11), (8,2), (2,7)\}$$

1. *Give the height of the tree*
2. *How many leafs there are?*
3. *The second leaf (starting from left) is the point with first coordinate the number ...*

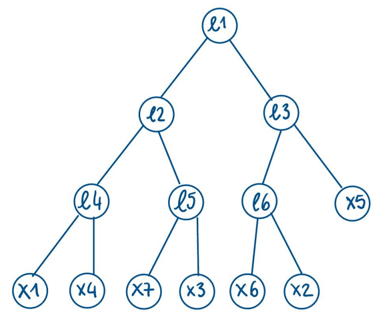
**Solution:** We denote by

$$x1 = (1,3), x2 = (12,1), \ldots, x7 = (2,7)$$

and we consider the splitting lines as defined in the BUILDKDTREE algorithm




The Kd-Tree in the output is the following one




Thus the tree has height 3, 7 leaves (i.e. the number of points) and the second left leave is the point $x4$.

**Exercise 2 ((It is enough to give an intuitive idea))** *Prove that the BUILD-KDTREE for a set of $n$ points has running time $O(n \log n)$ and uses $O(n)$ storage*

**Solution: (Here we give a proper proof)** Each splitting line divide the set of points in two equal parts (up to a unity). Therefore we divide the sets until lines divide two points in two parts. Therefore if the number of points is $n = 2^k$ then we need $2^k - 1$ lines (which are the parents in the tree, i.e. the internal nodes)

Indeed for splitting 2

, therefore the number of nodes (parents and leaves) is $2^k + 2^{k-1} = n + n/2 = 3n/2 < 3 \cdot n$.

If $n$ is not a power of 2, then there exists $t$ such that $2^{t-1} < n < 2^t$. Therefore the number $n_p$ of internal node (i.e. parents) is bounded by $2^{t-2} < n_p < 2^{t-1}$ (see above justification in the case of power of 2). Therefore the number of nodes (parents+leafs) satisfies

$$3 \cdot 2^{t-2} < n + n_p < 3 \cdot 2^{t-1}$$

therefore we have $n + n_p < 3 \cdot 2^{t-1} < 3 \cdot n$.

This show that the number of nodes is a $O(n)$. Now the statement about the storage is verified because each node uses $O(1)$ storage, so in total we need $O(1) \cdot O(n) = O(n)$ storage.

Now we consider the running time of BUILDKDTREE. It is a recursive process, where at each recursion a set of $n$ points is divided in two subsets of $n/2$ elements each (up to a unity). The cost of the split is linear in $n$ because we have to find the median, either with respect the $x$–coordinate or with respect the $y$–coordinate. Thus the building time $T(n)$ satisfies the recurrence

$$T(n) = \begin{cases} O(1) & \text{if } n = 1 \\ 2T(n/2) + O(n) & \text{if } n > 1 \end{cases}$$

Now we can apply Mater Theorem, as done in the case of Merge–Sort, and obtain that $T(n)$ is $O(n \log n)$.

**Exercise 3 (Optional)** *In Lecture 10 we have seen the recursive formula for the expected running time $E[T(n)]$ for Randomized–Select*

$$E[T(n)] = \frac{2}{n} \sum_{k=\lceil n/2 \rceil}^{n-1} \left( E[T(k)] + O(n) \right).$$

*Prove by induction that $E[T(n)] = O(n)$.*

Solution: The calculations are written at page 218 and page 219 of the book "Introduction to Algorithms" by Cormen et al.

**Exercise 4 (Optional)** *In Lecture 10 we have seen the recursive formula for the running time $T(n)$ for SELECT:*

$$T(n) \leq \begin{cases} = O(1) & \textit{if } n < 140 \\ T(\lceil n/5) + T(7n/10 + 6) + O(n) & \textit{if } n \geq 140 \end{cases}$$

*Prove by induction that the running time $T(n)$ is in $O(n)$*

Solution: The calculations are written at page 222 of the book "Introduction to Algorithms" by Cormen et al.