Solution to Exercise 4.3 (d)

Let \( T_I \) denote the initial set \( T \) and \( n := |T_I| \). For \( z \in \mathbb{R}_{>0} \) let \( T_z := \{ t \in T : ||t - r||_1 \geq z \} \). We construct the tree in a sweep-line manner starting with \( z = \max \{ ||t - r||_1 : t \in T_I \} + 1 \) and ending with \( z = 0 \). At each step of the algorithm we try to include Steiner points which have distance \( z \) from \( r \).

During the whole algorithm we store \( T_z \) in an array sorted circularly as follows:

1. elements \( p \) s.t. \( p - r \) lies in the 1st quadrant ordered by \( x \)-coord. in non-decr. order
2. elements \( p \) s.t. \( p - r \) lies in the 2nd quadrant ordered by \( x \)-coord. in non-incr. order
3. elements \( p \) s.t. \( p - r \) lies in the 3rd quadrant ordered by \( x \)-coord. in non-incr. order
4. elements \( p \) s.t. \( p - r \) lies in the 4th quadrant ordered by \( x \)-coord. in non-decr. order

We say that two points \( t_1, t_2 \in T_z \) are neighbors, if they are adjacent in \( T_z \) w.r.t. that ordering (or if one of them is the first and the other is the last element).

We store the set \( S_z := (T \setminus T_z) \cup \{ \text{med}(t_1, t_2, r) : t_1, t_2 \in T_z \text{ neighbors} \} \) in a 2-heap with key \( ||. - r||_1 \) such that we can determine the element with maximum key very fast. By convention, we say that in case of ties, \textsc{deletemax} returns elements of \( T \setminus T_z \) first.

\textbf{Claim:} If \( t_1, t_2 \in T \) s.t. \( ||\text{med}(t_1, t_2, r) - r||_1 = z \), then there are neighbors \( t'_1, t'_2 \in T_z \) such that \( ||\text{med}(t'_1, t'_2, r) - r||_1 \geq z \).

\textbf{Proof:} Assume that the claim is wrong. We select a counter-example \( t_1, t_2 \) such that \( t_1 \) and \( t_2 \) are non-neighbors which are as close as possible w.r.t. the ordering of \( T_z \). W.l.o.g., we may assume that \( r = (0, 0) \) and \( t_1 \) is in the 1st quadrant. If \( t_2 \) is in the 3rd quadrant, \( \text{med}(t_1, t_2, r) = r \) and the claim is trivial. Let \( t_2 \) be in the 2nd quadrant. We have: \( \text{med}(t_1, t_2, r) = (\min\{x(t_1), x(t_2)\}, 0) = (z, 0) \). Let \( t \in T_z \) come after \( t_1 \) but before \( t_2 \) in the ordering of \( T_z \). Since \( x(t_1) \leq x(t) \) if \( t \) is in the first quadrant and \( x(t_2) \leq x(t) \) if \( t \) is in the second quadrant, it holds that \( \min\{x(t_1), x(t_2)\} \leq \min\{x(t_1), x(t)\} \) and hence, \( ||\text{med}(t_1, t, r) - r||_1 \geq ||\text{med}(t_1, t_2, r) - r||_1 = z \).

Let \( t_2 \) be in the 4th quadrant. We have: \( \text{med}(t_1, t_2, r) = (0, \min\{y(t_1), y(t_2)\}) = (0, z) \). Since \( t_1 \) and \( t_2 \) are non-neighbors, \( t_1 \) is not the first or \( t_4 \) is not the last element of \( T_z \). W.l.o.g. let the first case be true (the other case is similar). Let \( t \) be the first element of \( T_z \). If \( y(t) \geq z \), we are done. Otherwise, \( 0 \leq y(t) < z \leq y(t_1) \) and thus, \( \text{med}(t, t_1, r) = t \) which concludes the case. Finally, assume that both \( t_1 \) and \( t_2 \) are in the first quadrant, w.l.o.g. \( x(t_1) \leq x(t_2) \). Let \( t \) be any vertex between \( t_1 \) and \( t_2 \). If \( y(t_1) \leq y(t) \) or \( y(t) \leq y(t_2) \), we are done. Otherwise, \( y(t_1) > y(t) > y(t_2) \) and \( ||\text{med}(t_1, t, r) - r||_1 = x(t_1) + y(t) \geq x(t_1) + y(t_2) = ||\text{med}(t_1, t_2, r) - r||_1 = z \). □
Assume, we have already computed $T_z$ and $S_z$ for some $z > 0$ and that we have already included all Steiner points with distance larger than $z$ from $r$. Let $t_1, t_2 \in T$ such that $||\text{med}(t_1, t_2, r) - r||_1$ is maximum. There are two cases:

**Case (i):** $||\text{med}(t_1, t_2, r) - r||_1 < z$. Then, we cannot include a Steiner point with distance $z$ from $r$. All further Steiner points have distance at least $z' := \max\{|s - r|_1 : s \in S_z\}$ which is exactly the maximum key of the heap. We can decrease $z$ to $z'$, include all sinks with distance $z'$ to the root to $T_z$ and update $S_z$ appropriately. Let $T'$ be the set of new elements in $T_z$. The number of elements we have to include to $S_z$ (i.e. points which arise by joining an element of $T'$ with a neighbor) as well as the points which we have to remove from $S_z$ (i.e. $T'$ and points which arise by joining elements in $T_z$ which are no longer neighbors) is at most $c \cdot |T'|$ for a constant $c$.

We can determine $T'$ by $|T'| \text{ DELETEMAX}$ operations. We can determine the correct position of a new element in an ordered list in $O(\log(|T_z|))$ time. Deleting from or inserting into a 2-heap takes $O(\log(n))$ time. Hence, this iteration can be performed in $O(|T'| \cdot \log(n))$ time.

**Case (ii):** $||\text{med}(t_1, t_2, r) - r||_1 = z$. By the claim, $t_1$ and $t_2$ can be chosen to be neighbors in $T_z$ and hence, $\text{med}(t_1, t_2, r)$ is an element of $S_z$ with maximum key. The key of each sink in $S_z$ is strictly smaller than $z$.

Hence, we can determine $t' := \text{med}(t_1, t_2, r)$ (or a median-point with the same distance to $r$) by a $\text{DELETEMAX}$ operation. We include $t'$ to $T_z$ but delete $t_1$ and $t_2$. We delete all (at most 3) points arising by joining $t_1$ or $t_2$ with a neighbor from $S_z$ ($t'$ is among them) and include all (at most 2) points arising by joining $t'$ with a neighbor. All of these operations need $O(\log(n))$ time.

When the heap is empty, $|T_z| = 1$ and we can connect the remaining sink with $r$.

Since all of the sets $T'$ as defined in case (i) are pairwise disjoint subsets of $T_I$, all case (i)-operations in total take $O(n \cdot \log(n))$ time.

All applications of case (ii) result in the insertion of a Steiner point. Let $S$ be the set of Steiner points. By construction, $|\delta(s)| = 3$ for each $s \in S$ and $|\delta(t)| = 1$ for $t \in T_I \cup \{r\}$. Thus,

$$|(S| + n + 1) - 1 = \frac{1}{2} \sum_{v \in S \cup T_I \cup \{r\}} |\delta(v)| = \frac{1}{2} \cdot (3|S| + n + 1)$$

which implies $|S| = n - 1$. Hence, all case (ii)-operations in total require $O(n \cdot \log(n))$ time.