Chapter 7

Voronoi Diagrams

7.1 Post Office Problem

Suppose there are n post offices $p_1, \ldots p_n$ in a city. Someone who is located at a position q within the city would like to know which post office is closest to him.[1] Modeling the city as a planar region, we think of $p_1, \ldots p_n$ and q as points in the plane. Denote the set of post offices by $P = \{p_1, \ldots p_n\}$.

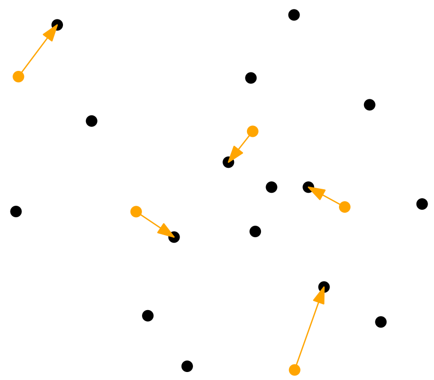

Figure 7.1: *Closest post offices for various query points.*

While the locations of post offices are known and do not change so frequently, we do not know in advance for which—possibly many—query locations the closest post office is to be found. Therefore, our long term goal is to come up with a data structure on top of P that allows to answer any possible query efficiently. The basic idea is to apply a so-called *locus approach*: we partition the query space into regions on which is the answer is the same. In our case, this amounts to partition the plane into regions such that for all points within a region the same point from P is closest (among all points from P).

[1] Another—possibly historically more accurate—way to think of the problem: You want to send a letter to a person living at q. For this you need to know the corresponding zip code, which is the code of the post office closest to q.

As a warmup, consider the problem for two post offices $p_i, p_j \in P$. For which query locations is the answer p_i rather than p_j? This region is bounded by the bisector of p_i and p_j, that is, the set of points which have the same distance to both points.

Proposition 7.1. *For any two distinct points in $\mathbb{R}^d$ the bisector is a hyperplane, that is, in $\mathbb{R}^2$ it is a line.*

Proof. Let $p = (p_1, \ldots, p_d)$ and $q = (q_1, \ldots, q_d)$ be two points in $\mathbb{R}^d$. The bisector of p and q consists of those points $x = (x_1, \ldots, x_d)$ for which

$$\begin{aligned}
\|p - x\| = \|q - x\| &\iff \|p - x\|^2 = \|q - x\|^2 \\
&\iff \sum_{i=1}^{d} (p_i - x_i)^2 = \sum_{i=1}^{d} (q_i - x_i)^2 \\
&\iff \sum_{i=1}^{d} {p_i}^2 - 2\sum_{i=1}^{d} p_i x_i + \sum_{i=1}^{d} {x_i}^2 = \sum_{i=1}^{d} {q_i}^2 - 2\sum_{i=1}^{d} q_i x_i + \sum_{i=1}^{d} {x_i}^2 \\
&\iff \sum_{i=1}^{d} {p_i}^2 - \sum_{i=1}^{d} {q_i}^2 = 2\sum_{i=1}^{d} (p_i - q_i) x_i \\
&\iff \|p\|^2 - \|q\|^2 = 2(p - q)^\top x\,.
\end{aligned}$$

As p and q are distinct, this is the equation of a hyperplane. □

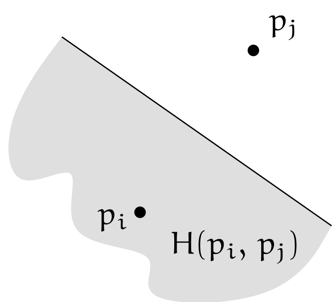

Figure 7.2: *The bisector of two points.*

Denote by $H(p_i, p_j)$ the closed halfspace bounded by the bisector of p_i and p_j that contains p_i. In $\mathbb{R}^2$, the region $H(p_i, p_j)$ is a halfplane; see Figure 7.2.

Exercise 7.2.

a) *What is the bisector of a line ℓ and a point $p \in \mathbb{R}^2 \setminus \ell$, that is, the set of all points $x \in \mathbb{R}^2$ with $\|x - p\| = \|x - \ell\| = \min_{q \in \ell} \|x - q\|$?*

b) *For two points $p \neq q \in \mathbb{R}^2$, what is the region that contains all points whose distance to p is exactly twice their distance to q?*

7.2 Voronoi Diagram

As it turns out, understanding the situation for two points essentially tells us everything we need to know for the general case. The structure obtained by applying the locus approach to the nearest neighbor problem is called *Voronoi diagram*. In fact, this approach works for a variety of distance functions and spaces [2, 7]. So, Voronoi diagram should be considered a family of structures rather than a single specific one. Without further qualification, the underlying distance function is the Euclidean metric. In the following we define and study the Voronoi diagram for a given set $P = \{p_1, \ldots, p_n\}$ of points in $\mathbb{R}^2$.

Definition 7.3. *For* $p_i \in P$ *denote the* **Voronoi cell** $V_P(i)$ *of* p_i *by*

$$V_P(i) := \left\{q \in \mathbb{R}^2 : \|q - p_i\| \leqslant \|q - p\| \text{ for all } p \in P\right\}.$$

Proposition 7.4.

$$V_P(i) = \bigcap_{j \neq i} H(p_i, p_j).$$

Proof. For $j \neq i$ we have $\|q - p_i\| \leqslant \|q - p_j\| \iff q \in H(p_i, p_j)$. □

Corollary 7.5. $V_P(i)$ *is non-empty and convex.*

Proof. According to Proposition 7.4, the region $V_P(i)$ is the intersection of a finite number of halfplanes and hence convex. As $p_i \in V_P(i)$, we have $V_P(i) \neq \emptyset$. □

Observe that every point of the plane lies in some Voronoi cell but no point lies in the interior of two Voronoi cells. Therefore these cells form a *subdivision* of the plane (a partition[2] into interior-disjoint simple polygons). See Figure 7.3 for an example.

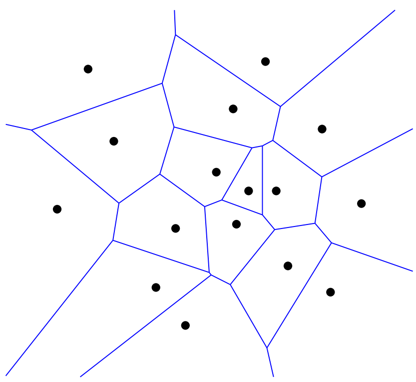

Figure 7.3: *Example: The Voronoi diagram of a point set.*

[2]Strictly speaking, to obtain a partition, we treat the shared boundaries of the polygons as separate entities.

Definition 7.6. *The* **Voronoi Diagram** $\mathrm{VD}(P)$ *of a set* $P = \{p_1, \ldots, p_n\}$ *of points in* $\mathbb{R}^2$ *is the subdivision of the plane induced by the Voronoi cells* $V_P(i)$, *for* $i = 1, \ldots, n$. *Denote by* $\mathrm{VV}(P)$ *the set of vertices, by* $\mathrm{VE}(P)$ *the set of edges, and by* $\mathrm{VR}(P)$ *the set of regions (faces) of* $\mathrm{VD}(P)$.

Lemma 7.7. *For every vertex* $v \in \mathrm{VV}(P)$ *the following statements hold.*

a) v *is the common intersection of at least three edges from* $\mathrm{VE}(P)$;

b) v *is incident to at least three regions from* $\mathrm{VR}(P)$;

c) v *is the center of a circle* $C(v)$ *through at least three points from* P *such that*

d) $D(v)^\circ \cap P = \emptyset$, *where* $D(v)$ *denotes the disk bounded by* $C(v)$.

Proof. Consider a vertex $v \in \mathrm{VV}(P)$. As all Voronoi cells are convex, $k \geqslant 3$ of them must be incident to v. This proves Part a) and b).

Without loss of generality let these cells be $V_P(i)$, for $1 \leqslant i \leqslant k$; see Figure 7.4. Denote by e_i, $1 \leqslant i \leqslant k$, the edge incident to v that bounds $V_P(i)$ and $V_P((i \bmod k)+1)$.

For any $i = 1, \ldots, k$ we have $v \in e_i \Rightarrow \|v - p_i\| = \|v - p_{(i \bmod k)+1}\|$. In other words, $p_1, p_2, \ldots, p_k$ are cocircular, which proves Part c).

Part d): Suppose there exists a point $p_\ell \in D(v)^\circ$. Then the vertex v is closer to p_ℓ than it is to any of $p_1, \ldots, p_k$, in contradiction to the fact that v is contained in all of $V_P(1), \ldots, V_P(k)$. □

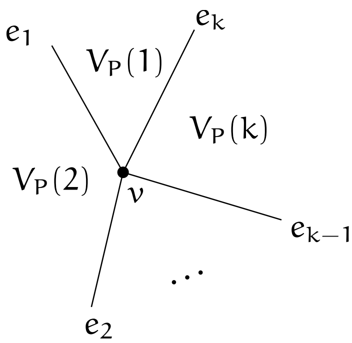

Figure 7.4: *Voronoi regions around* v.

Corollary 7.8. *If* P *is in general position (no four points from* P *are cocircular), then for every vertex* $v \in \mathrm{VV}(P)$ *the following statements hold.*

a) v *is the common intersection of exactly three edges from* $\mathrm{VE}(P)$;

b) v *is incident to exactly three regions from* $\mathrm{VR}(P)$;

c) v *is the center of a circle* $C(v)$ *through exactly three points from* P *such that* *d)* $D(v)^\circ \cap P = \emptyset$*, where* $D(v)$ *denotes the disk bounded by* $C(v)$*.* □

Lemma 7.9. *There is an unbounded Voronoi edge bounding* $V_P(i)$ *and* $V_P(j)$ $\iff$ $\overline{p_ip_j} \cap P = \{p_i, p_j\}$ *and* $\overline{p_ip_j} \subseteq \partial\mathrm{conv}(P)$*, where the latter denotes the boundary of the convex hull of* P*.*

Proof. Denote by $b_{i,j}$ the bisector of p_i and p_j, and let $\mathcal{D}$ denote the family of disks centered at some point on $b_{i,j}$ and passing through p_i (and p_j). There is an unbounded Voronoi edge bounding $V_P(i)$ and $V_P(j)$ $\iff$ there is a ray $\rho \subset b_{i,j}$ such that $\|r-p_k\| > \|r-p_i\|$ $(= \|r-p_j\|)$, for every $r \in \rho$ and every $p_k \in P$ with $k \notin \{i,j\}$. Equivalently, there is a ray $\rho \subset b_{i,j}$ such that for every point $r \in \rho$ the disk $C \in \mathcal{D}$ centered at r does not contain any point from P in its interior (Figure 7.5).

The latter statement implies that the open halfplane H, whose bounding line passes through p_i and p_j and such that H contains the infinite part of ρ, contains no point from P in its interior. Therefore, $\overline{p_ip_j}$ appears on $\partial\mathrm{conv}(P)$ and $\overline{p_ip_j}$ does not contain any $p_k \in P$, for $k \neq i,j$.

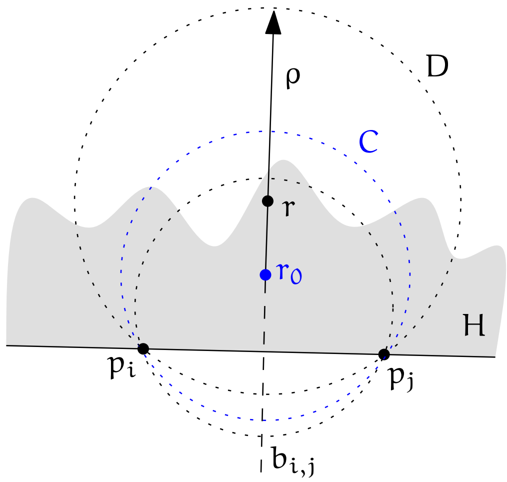

Figure 7.5: *The correspondence between* $\overline{p_ip_j}$ *appearing on* $\partial\mathrm{conv}(P)$ *and a family* $\mathcal{D}$ *of empty disks centered at the bisector of* p_i *and* p_j*.*

Conversely, suppose that $\overline{p_ip_j}$ appears on $\partial\mathrm{conv}(P)$ and $\overline{p_ip_j} \cap P = \{p_i, p_j\}$. Then some halfplane H whose bounding line passes through p_i and p_j contains no point from P in its interior. In particular, the existence of H together with $\overline{p_ip_j} \cap P = \{p_i, p_j\}$ implies that there is some disk $C \in \mathcal{D}$ such that $C \cap P = \{p_i, p_j\}$. Denote by r_0 the center of C and let ρ denote the ray starting from r_0 along $b_{i,j}$ such that the infinite part of ρ is contained in H. Consider any disk $D \in \mathcal{D}$ centered at a point $r \in \rho$ and observe that $D \setminus H \subseteq C \setminus H$. As neither H nor C contain any point from P in their respective interior, neither does D. This holds for every D, and we have seen above that this statement is equivalent to the existence of an unbounded Voronoi edge bounding $V_P(i)$ and $V_P(j)$. □

7.3 Duality

A *straight-line dual* of a plane graph G is a graph G' defined as follows: Choose a point for each face of G and connect any two such points by a straight edge, if the corresponding faces share an edge of G. Observe that this notion depends on the embedding; that is why the straight-line dual is defined for a plane graph rather than for an abstract graph. In general, G' may have edge crossings, which may also depend on the choice of representative points within the faces. However, for Voronoi diagrams there is a particularly natural choice of representative points such that G' is plane: the points from P.

Theorem 7.10 (Delaunay [3]). *The straight-line dual of* $\mathrm{VD}(P)$ *for a set* $P \subset \mathbb{R}^2$ *of* $n \geqslant 3$ *points in general position (no three points from* P *are collinear and no four points from* P *are cocircular) is a triangulation: the unique Delaunay triangulation of* P.

Proof. By Lemma 7.9, the convex hull edges appear in the straight-line dual T of $\mathrm{VD}(P)$ and they correspond exactly to the unbounded edges of $\mathrm{VD}(P)$. All remaining edges of $\mathrm{VD}(P)$ are bounded, that is, both endpoints are Voronoi vertices. Consider some $v \in \mathrm{VV}(P)$. According to Corollary 7.8(b), v is incident to exactly three Voronoi regions, which, therefore, form a triangle $\triangle(v)$ in T. By Corollary 7.8(d), the circumcircle of $\triangle(v)$ does not contain any point from P in its interior. Hence $\triangle(v)$ appears in the (unique by Corollary 5.17) Delaunay triangulation of P.

Conversely, for any triangle $p_ip_jp_k$ in the Delaunay triangulation of P, by the empty circle property the circumcenter c of $p_ip_jp_k$ has p_i, p_j, and p_k as its closest points from P. Therefore, $c \in \mathrm{VV}(P)$ and—as above—the triangle $p_ip_jp_k$ appears in T. □

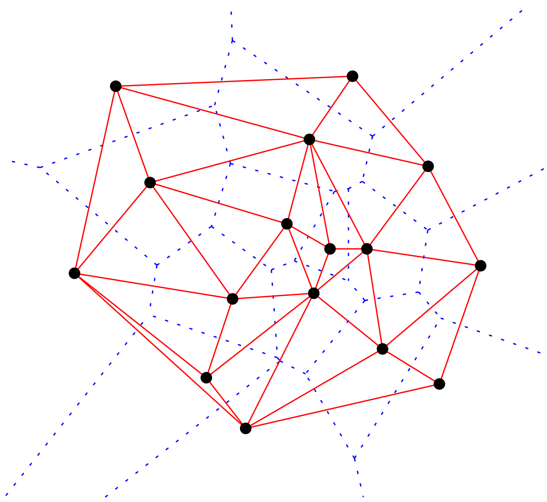

Figure 7.6: *The Voronoi diagram of a point set and its dual Delaunay triangulation.*

It is not hard to generalize Theorem 7.10 to general point sets. In this case, a Voronoi vertex of degree k is mapped to a convex polygon with k cocircular vertices. Any triangulation of such a polygon yields a Delaunay triangulation of the point set.

Corollary 7.11. $|VE(P)| \leqslant 3n - 6$ *and* $|VV(P)| \leqslant 2n - 5$.

Proof. Every edge in $VE(P)$ corresponds to an edge in the dual Delaunay triangulation. The latter is a plane graph on n vertices, which by Corollary 2.5 has at most $3n-6$ edges and at most $2n-4$ faces. Only the bounded faces correspond to a vertex in $VD(P)$. □

Corollary 7.12. *For a set $P \subset \mathbb{R}^2$ of n points, the Voronoi diagram of P can be constructed in expected $O(n \log n)$ time and $O(n)$ space.*

Proof. We have seen that a Delaunay triangulation T for P can be obtained using randomized incremental construction in the given time and space bounds. As T is a plane graph, its number of vertices, edges, and faces all are linear in n. Therefore, the straight-line dual of T—which by Theorem 7.10 is the desired Voronoi diagram—can be computed in $O(n)$ additional time and space. □

Exercise 7.13. *Consider the Delaunay triangulation T for a set $P \subset \mathbb{R}^2$ of $n \geqslant 3$ points in general position. Prove or disprove:*

a) Every edge of T intersects its dual Voronoi edge.

b) Every vertex of $VD(P)$ is contained in its dual Delaunay triangle.

7.4 Lifting Map

Recall the lifting map that we used in Section 5.3 to prove that the Lawson Flip Algorithm terminates. Denote by $\mathcal{U}\colon z = x^2 + y^2$ the unit paraboloid in $\mathbb{R}^3$. The lifting map $\ell\colon \mathbb{R}^2 \to \mathcal{U}$ with $\ell\colon p = (p_x, p_y) \mapsto (p_x, p_y, {p_x}^2 + {p_y}^2)$ is the projection of the x/y-plane onto $\mathcal{U}$ in direction of the z-axis.

For $p \in \mathbb{R}^2$ let H_p denote the plane of tangency to $\mathcal{U}$ in $\ell(p)$. Denote by $h_p\colon \mathbb{R}^2 \to H_p$ the projection of the x/y-plane onto H_p in direction of the z-axis (see Figure 7.7).

Lemma 7.14. $\|\ell(q) - h_p(q)\| = \|p - q\|^2$, *for any points $p, q \in \mathbb{R}^2$.*

Exercise 7.15. *Prove Lemma 7.14.* Hint: *First determine the equation of the tangent plane H_p to $\mathcal{U}$ in $\ell(p)$.*

Theorem 7.16. *For $p = (p_x, p_y) \in \mathbb{R}^2$ denote by H_p the plane of tangency to the unit paraboloid $\mathcal{U} = \{(x, y, z) : z = x^2 + y^2\} \subset \mathbb{R}^3$ in $\ell(p) = (p_x, p_y, {p_x}^2 + {p_y}^2)$. Let $\mathcal{H}(P) := \bigcap_{p \in P} H_p^+$ the intersection of all halfspaces above the planes H_p, for $p \in P$. Then the vertical projection of $\partial\mathcal{H}(P)$ onto the x/y-plane forms the Voronoi Diagram of P (the faces of $\partial\mathcal{H}(P)$ correspond to Voronoi regions, the edges to Voronoi edges, and the vertices to Voronoi vertices).*

Proof. For any point $q \in \mathbb{R}^2$, the vertical line through q intersects every plane H_p, $p \in P$. By Lemma 7.14 the topmost plane intersected belongs to the point from P that is closest to q. □

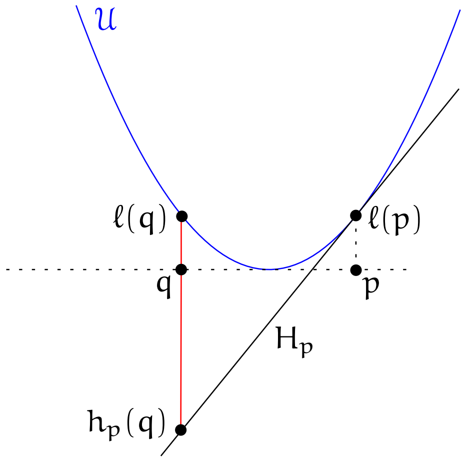

Figure 7.7: *Lifting map interpretation of the Voronoi diagram in a two-dimensional projection.*

7.5 Planar Point Location

One last bit is still missing in order to solve the post office problem optimally.

Theorem 7.17. *Given a triangulation* T *for a set* $P \subset \mathbb{R}^2$ *of* n *points, one can build in* $O(n)$ *time an* $O(n)$ *size data structure that allows for any query point* $q \in \operatorname{conv}(P)$ *to find in* $O(\log n)$ *time a triangle from* T *containing* q*.*

The data structure we will employ is known as *Kirkpatrick's hierarchy*. But before discussing it in detail, let us put things together in terms of the post office problem.

Corollary 7.18 (Nearest Neighbor Search). *Given a set* $P \subset \mathbb{R}^2$ *of* n *points, one can build in expected* $O(n \log n)$ *time an* $O(n)$ *size data structure that allows for any query point* $q \in \operatorname{conv}(P)$ *to find in* $O(\log n)$ *time a nearest neighbor of* q *among the points from* P*.*

Proof. First construct the Voronoi Diagram V of P in expected $O(n \log n)$ time. It has exactly n convex faces. Every unbounded face can be cut by the convex hull boundary into a bounded and an unbounded part. As we are concerned with query points within $\operatorname{conv}(P)$ only, we can restrict our attention to the bounded parts.[3] Any convex polygon can easily be triangulated in time linear in its number of edges (= number of vertices). As V has at most $3n - 6$ edges and every edge appears in exactly two faces, V can be triangulated in $O(n)$ time overall. Label each of the resulting triangles with the point from p, whose Voronoi region contains it, and apply the data structure from Theorem 7.17. □

[3]We even know how to decide in $O(\log n)$ time whether or not a given point lies within $\operatorname{conv}(P)$, see Exercise 4.25.

7.6 Kirkpatrick's Hierarchy

We will now develop a data structure for point location in a triangulation, as described in Theorem 7.17. For simplicity we assume that the triangulation T we work with is a maximal planar graph, that is, the outer face is a triangle as well. This can easily be achieved by an initial normalization step that puts a huge triangle T_h around T and triangulates the region in between T_h and T (in linear time—how?).

The main idea for the data structure is to construct a hierarchy $T_0, \ldots, T_h$ of triangulations, such that

- $T_0 = T$,
- the vertices of T_i are a subset of the vertices of T_{i-1}, for $i = 1, \ldots, h$, and
- T_h is a single triangle only.

Search. For a query point x we can find a triangle from T that contains x as follows.

Search$(x \in \mathbb{R}^2)$

1. For $i = h, h-1, \ldots, 0$: Find a triangle t_i from T_i that contains x.
2. return t_0.

This search is efficient under the following conditions.

(C1) Every triangle from T_i intersects only few ($\leqslant c$) triangles from T_{i-1}. (These will then be connected via the data structure.)

(C2) h is small ($\leqslant d \log n$).

Proposition 7.19. *The search procedure described above needs* $\leqslant 3cd \log n = O(\log n)$ *orientation tests.*

Proof. For every T_i, $0 \leqslant i < h$, at most c triangles are tested as to whether or not they contain x. Using three orientation tests one can determine whether or not a triangle contains a given point. □

Thinning. Removing a vertex v and all its incident edges from a triangulation creates a non-triangulated hole that forms a star-shaped polygon since all points are visible from v (the star-point). Here we remove vertices of constant degree only and therefore these polygons are of constant size. But even if they were not, it is not hard to triangulate a star-shaped polygon in linear time.

Lemma 7.20. *A star-shaped polygon, given as a sequence of* $n \geqslant 3$ *vertices and a star-point, can be triangulated in* $O(n)$ *time.*

Exercise 7.21. *Prove Lemma 7.20.*

As a side remark, the *kernel* of a simple polygon, that is, the (possibly empty) set of all star-points, can be constructed in linear time as well using linear programming.

Our working plan is to obtain T_i from T_{i-1} by removing several *independent* (pairwise non-adjacent) vertices and re-triangulating. These vertices should

a) have small degree (otherwise the degree within the hierarchy gets too large, that is, we need to test too many triangles on the next level) and

b) be many (otherwise the height h of the hierarchy gets too large).

The following lemma asserts the existence of a sufficiently large set of independent small-degree vertices in every triangulation.

Lemma 7.22. *In every triangulation of n points in $\mathbb{R}^2$ there exists an independent set of at least $\lceil n/18 \rceil$ vertices of maximum degree 8. Moreover, such a set can be found in $O(n)$ time.*

Proof. Let $T = (V, E)$ denote the graph of the triangulation, which we consider as an abstract graph in the following. We may suppose that T is maximal planar, that is, the outer face is a triangle. (Otherwise use Theorem 2.26 to combinatorially triangulate T arbitrarily. An independent set in the resulting graph T' is also independent in T and the degree of a vertex in T' is at least as large as its degree in T.) For $n = 3$ the statement is true. Let $n \geqslant 4$.

By the Euler formula we have $|E| = 3n - 6$, that is,

$$\sum_{v \in V} \deg_T(v) = 2|E| = 6n - 12 < 6n.$$

Let $W \subseteq V$ denote the set of vertices of degree at most 8. Claim: $|W| > n/2$. Suppose $|W| \leqslant n/2$. By Theorem 2.27 we know that T is 3-connected and so every vertex has degree at least three. Therefore

$$\begin{aligned} \sum_{v \in V} \deg_T(v) &= \sum_{v \in W} \deg_T(v) + \sum_{v \in V \setminus W} \deg_T(v) \geqslant 3|W| + 9|V \setminus W| \\ &= 3|W| + 9(n - |W|) = 9n - 6|W| \geqslant 9n - 3n = 6n, \end{aligned}$$

in contradiction to the above.

Construct an independent set U in T as follows (greedily): As long as $W \neq \emptyset$, add an arbitrary vertex $v \in W$ to U and remove v and all its neighbors from W. Assuming that T is represented so that we can obtain the neighborhood of a vertex v in $\deg_T(v)$ time (for instance, using adjacency lists), both W and U can be computed in $O(n)$ time.

Obviously U is independent and all vertices in U have degree at most 8. At each selection step at most 9 vertices are removed from W. Therefore $|U| \geqslant \lceil (n/2)/9 \rceil = \lceil n/18 \rceil$. □

Proof. (of Theorem 7.17)
Construct the hierarchy $T_0, \ldots T_h$ with $T_0 = T$ as follows. Obtain T_i from T_{i-1} by removing an independent set U as in Lemma 7.22 and re-triangulating the resulting holes. By Lemma 7.20 and Lemma 7.22 every step is linear in the number $|T_i|$ of vertices in T_i. The total cost for building the data structure is thus

$$\sum_{i=0}^{h} \alpha|T_i| \leqslant \sum_{i=0}^{h} \alpha n \left(1 - \frac{1}{18}\right)^i \leqslant \sum_{i=0}^{h} \alpha n (17/18)^i < \alpha n \sum_{i=0}^{\infty} (17/18)^i = 18\alpha n \in O(n),$$

for some constant α. Similarly the space consumption is linear.

The number of levels amounts to $h = \log_{18/17} n < 12.2 \log n$. Thus by Proposition 7.19 the search needs at most $3 \cdot 8 \cdot \log_{18/17} n < 292 \log n$ orientation tests. □

Improvements. As the name suggests, the hierarchical approach discussed above is due to David Kirkpatrick [6]. The constant 292 that appears in the search time is somewhat large. There has been a whole line of research trying to improve it using different techniques.

- Sarnak and Tarjan [8]: $4 \log n$.
- Edelsbrunner, Guibas, and Stolfi [4]: $3 \log n$.
- Goodrich, Orletsky, and Ramaiyer [5]: $2 \log n$.
- Adamy and Seidel [1]: $1 \log n + 2\sqrt{\log n} + O(\sqrt[4]{\log n})$.

Exercise 7.23. *Let $\{p_1, p_2, \ldots, p_n\}$ be a set of points in the plane, which we call* obstacles*. Imagine there is a disk of radius r centered at the origin which can be moved around the obstacles but is not allowed to intersect them (touching the boundary is ok). Is it possible to move the disk out of these obstacles? See the example depicted in Figure 7.8 below.*

More formally, the question is whether there is a (continuous) path $\gamma : [0,1] \longrightarrow \mathbb{R}^2$ with $\gamma(0) = (0,0)$ and $\|\gamma(1)\| \geqslant \max\{\|p_1\|, \ldots, \|p_n\|\}$, such that at any time $t \in [0,1]$ and $\|\gamma(t) - p_i\| \geqslant r$, for any $1 \leqslant i \leqslant n$. Describe an algorithm to decide this question and to construct such a path—if one exists—given arbitrary points $\{p_1, p_2, \ldots, p_n\}$ and a radius $r > 0$. Argue why your algorithm is correct and analyze its running time.

Exercise 7.24. *This exercise is about an application from* Computational Biology*: You are given a set of disks $P = \{a_1, .., a_n\}$ in $\mathbb{R}^2$, all with the same radius $r_a > 0$. Each of these disks represents an atom of a protein. A water molecule is represented by a disc with radius $r_w > r_a$. A water molecule cannot intersect the interior of any protein atom, but it can be tangent to one. We say that an atom $a_i \in P$ is* accessible *if there exists a placement of a water molecule such that it is tangent to a_i and does not intersect the interior of any other atom in P. Given P, find an $O(n \log n)$ time algorithm which determines all atoms of P that are inaccessible.*

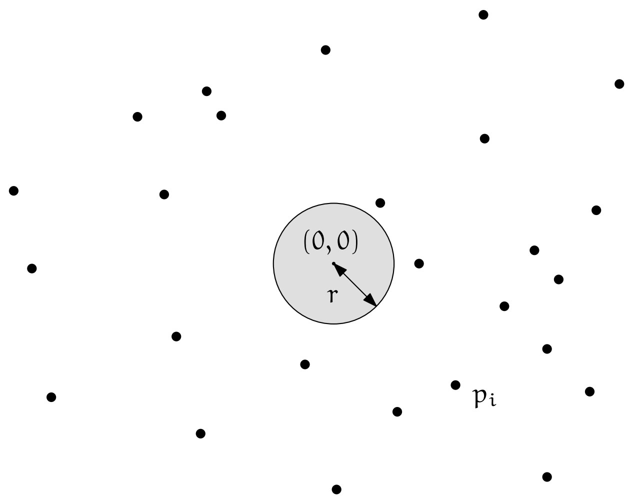

Figure 7.8: *Motion planning: Illustration for Exercise 7.23.*

Exercise 7.25. *Let $P \subset \mathbb{R}^2$ be a set of n points. Describe a data structure to find in $O(\log n)$ time a point in P that is furthest from a given query point q among all points in P.*

Exercise 7.26. *Show that the bounds given in Theorem 7.17 are optimal in the algebraic computation tree model.*

Questions

30. *What is the Voronoi diagram of a set of points in $\mathbb{R}^2$?* Give a precise definition and explain/prove the basic properties: convexity of cells, why is it a subdivision of the plane?, Lemma 7.7, Lemma 7.9.
31. *What is the correspondence between the Voronoi diagram and the Delaunay triangulation for a set of points in $\mathbb{R}^2$?* Prove duality (Theorem 7.10) and explain where general position is needed.
32. *How to construct the Voronoi diagram of a set of points in $\mathbb{R}^2$?* Describe an $O(n \log n)$ time algorithm, for instance, via Delaunay triangulation.
33. *How can the Voronoi diagram be interpreted in context of the lifting map?* Describe the transformation and prove its properties to obtain a formulation of the Voronoi diagram as an intersection of halfspaces one dimension higher.
34. *What is the Post-Office Problem and how can it be solved optimally?* Describe the problem and a solution using linear space, $O(n \log n)$ preprocessing, and $O(\log n)$ query time.
35. *How does Kirkpatrick's hierarchical data structure for planar point location work exactly?* Describe how to build it and how the search works, and prove the

runtime bounds. In particular, you should be able to state and prove Lemma 7.22 and Theorem 7.17.

References

[1] Udo Adamy and Raimund Seidel, On the exaxt worst case query complexity of planar point location. *J. Algorithms*, **37**, (2000), 189–217.

[2] Franz Aurenhammer, Voronoi diagrams: A survey of a fundamental geometric data structure. *ACM Comput. Surv.*, **23**, 3, (1991), 345–405.

[3] Boris Delaunay, Sur la sphère vide. A la memoire de Georges Voronoi. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskih i Estestvennyh Nauk*, **6**, (1934), 793–800.

[4] Herbert Edelsbrunner, Leonidas J. Guibas, and Jorge Stolfi, Optimal point location in a monotone subdivision. *SIAM J. Comput.*, **15**, 2, (1986), 317–340.

[5] Michael T. Goodrich, Mark W. Orletsky, and Kumar Ramaiyer, Methods for achieving fast query times in point location data structures. In *Proc. 8th ACM-SIAM Sympos. Discrete Algorithms*, pp. 757–766, 1997.

[6] David G. Kirkpatrick, Optimal search in planar subdivisions. *SIAM J. Comput.*, **12**, 1, (1983), 28–35.

[7] Rolf Klein, *Concrete and abstract Voronoi diagrams*, vol. 400 of *Lecture Notes Comput. Sci.*, Springer, 1989.

[8] Neil Sarnak and Robert E. Tarjan, Planar point location using persistent search trees. *Commun. ACM*, **29**, 7, (1986), 669–679.