Chapter 7

Incremental Construction of Delaunay Triangulation

We have learned about the Lawson flip algorithm which computes a Delaunay triangulation of a given n-point set $P \subseteq \mathbb{R}^2$ by performing $O(n^2)$ flips. With some care, the algorithm can be implemented to run in $O(n^2)$ time. On the other hand, we have also seen in an exercise that certain point sets require $\Omega(n^2)$ flips, meaning that the worst-case running time is $\Theta(n^2)$.

Here we will present a different, randomized algorithm which runs in $O(n \log n)$ time in expectation. (The probability comes from the random choices made by the algorithm, not from the input P.) Throughout we assume general position (no three points collinear and no four points cocircular), so that the Delaunay triangulation is unique by Corollary 6.18. There are techniques to deal with non-general position, but we will leave that out.

7.1 Incremental construction

To avoid special cases, we augment the set P with three "far-out" points a, b and c. For now suffice it to say that the huge triangle abc contains P with abundant space cushion.

The idea is to start from the triangle abc and insert other points one after another according to a uniformly random order $(p_1, p_2, \ldots, p_n)$ of P. For $1 \leqslant s \leqslant n$, we denote $P_s = \{p_1, \ldots, p_s\}$ and $P_s^+ = \{a, b, c\} \cup P_s$. Suppose that in the first $s-1$ rounds we had built the Delaunay triangulation $\mathcal{D}_{s-1}$ of P_{s-1}^+. At round s we shall insert point p_s and repair the structure to get the Delaunay triangulation $\mathcal{D}_s$ of P_s^+. In the end, we obtain the Delaunay triangulation $\mathcal{D}_n$ of P_n^+.

From $\mathcal{D}_n$ we want to "read off" the Delaunay triangulation of P by simply ignoring the three artificial points. For this to work, the convex hull boundary $\partial\mathrm{conv}(P)$ should be respected by $\mathcal{D}_n$. It can be ensured by placing a, b, c far enough so that they are not enclosed by the empty circumcircles going through adjacent convex hull vertices. But practically speaking, a simpler approach is to choose $a = (-\infty, -\infty)$, $b = (\infty, -\infty)$ and

$c = (0, \infty)$ and extend the algebra to handle symbols $-\infty, \infty$.

Below is the outline of round s, which will be fleshed out in subsequent sections. In our figures, we suppress the artificial points since they are merely a technicality.

(a) Find the triangle $\Delta \in \mathcal{D}_{s-1}$ that contains p_s, and split it into three triangles by connecting p_s with the three vertices of Δ. We now have a triangulation $\mathcal{T}$ of P_s^+. (Figure 7.1a)

(b) Perform Lawson flips on $\mathcal{T}$ until we obtain the Delaunay triangulation $\mathcal{D}_s$. (Figure 7.1b)

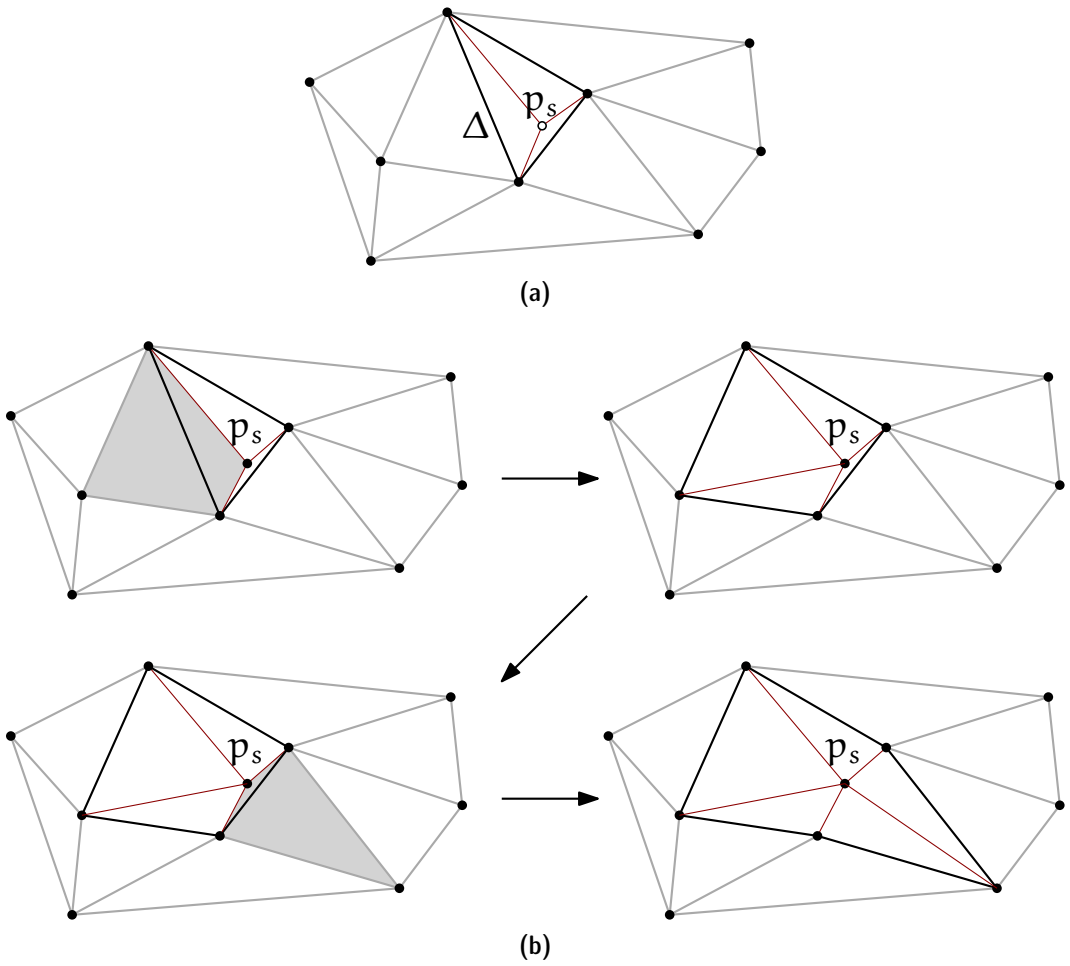

Figure 7.1: *Insert p_s to $\Delta \in \mathcal{D}_{s-1}$ and perform Lawson flips.*

7.2 Organizing the Lawson flips

First off, let us implement (b) in the outline. It turns out that the Lawson flips proceed quite systematically.

Lemma 7.1. *The following invariants hold at any particular moment in round s:*

(i) Every edge incident to p_s must belong to $\mathcal{D}_s$; in particular, it cannot be flipped away in the rest of round s.

(ii) Every applicable Lawson flip at this moment involves some triangle p_suv and some triangle $uvw \in \mathcal{D}_{s-1}$. It replaces them with triangles p_suw and p_svw, both incident to p_s, thus the degree of p_s increases by one.

Proof. We argue by strong induction over time. As the base case, we consider the moment before any flip is performed.

(i) Let us take any incident edge p_sw, where w must be a vertex of Δ. Since $\Delta \in \mathcal{D}_{s-1}$, its circumcircle C encloses nothing but the new point p_s. We can thus shrink C to an empty circle C' passing through p_s and w only, see Figure 7.2a. So the edge p_sw must be in $\mathcal{D}_s$ by Lemma 6.17.

(ii) Only the three edges of Δ are potentially flippable, since they are the only edges whose incident triangles have changed and form a convex quadrilateral. So any next flip must adhere to the claimed format.

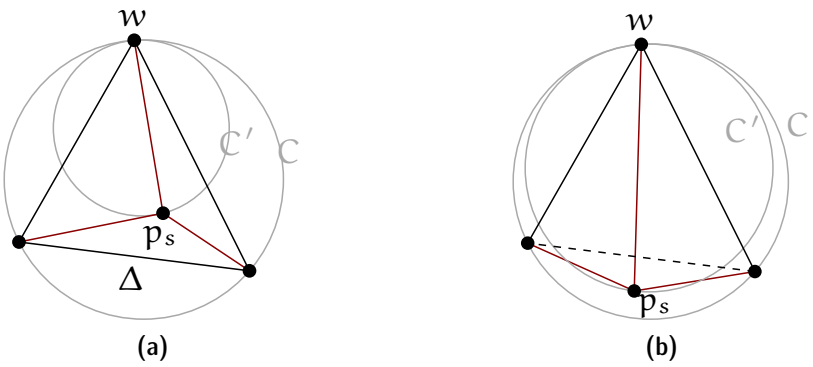

Figure 7.2: *Newly created edges incident to p_s are in the Delaunay graph*

Next we consider the moment after some flip(s) have been performed. Denote by $R \subseteq \mathbb{R}^2$ the union of all triangles incident to p_s right now. Note that R is a star-shaped polygon. One can see by inductively applying (ii) that the affected region of the previous flips is restricted in R. In other words, all triangles outside R are not yet touched, meaning they must belong to $\mathcal{D}_{s-1}$.

(i) Let us take the incident edge p_sw generated by the last flip. By induction hypothesis (ii), this flip destroys exactly one triangle in $\mathcal{D}_{s-1}$. Its circumcircle C contains p_s only, and shrinking it yields an empty circle C' through p_s and w, see Figure 7.2b. Thus p_sw must be in $\mathcal{D}_s$ by Lemma 6.17.

(ii) As established in (i), all edges incident to p_s are not flippable. So any flippable edges has to be a boundary edge of polygon R, say uv. On one side it is incident to some triangle $p_s uv$ (by definition of R); on the other side it is incident to a triangle $uvw \in \mathcal{D}_{s-1}$ (as we argued above).

This completes the induction. □

The lemma suggests that we can maintain a queue of potentially flippable edges that we process in turn. Initially the queue contains only the three edges of Δ. In each step, we remove an edge uv from the queue. If its two incident triangles $p_s uv$ and uvw are not locally Delaunay, then we perform the flip and push uw and vw to the queue. Otherwise we simply discard it because it cannot become flippable in the future. (Suppose to contradiction that it becomes flippable, then by Lemma 7.1 the flip must involve $p_s uv$ and some $uvw \in \mathcal{D}_{s-1}$. But the two triangles are in place right now, so we should have performed the flip right away.)

Corollary 7.2. *Let $d_s := \deg_{\mathcal{D}_s}(p_s)$ be the degree of vertex p_s in the (graph of) triangulation $\mathcal{D}_s$. Then in round s we perform exactly $d_s - 3$ Lawson flips. Moreover, these flips can be implemented to consume time only linear in d_s. The total number of triangles created in round s is $2d_s - 3$ (although some of them can be flipped away within the same round).*

Proof. By Lemma 7.1, every Lawson flip increases the number of edges incident to p_s by exactly one. So the number of flips is equal to the final degree d_s minus the initial degree 3. Each flip creates two new triangles, along with the initial three triangles we get $2d_s - 3$ in total. Using the queue implementation discussed above, every flip needs only a constant number of operations, so the total running time is linear in d_s. □

7.3 The History Graph

Let us get back to part (a) in the outline and specify how we find the triangle $\Delta \in \mathcal{D}_{s-1}$ that contains the point p_s. Doing this in the naïve way (checking all triangles) is not a good idea, as it would then amount to $\Theta(n^2)$ work throughout the whole algorithm. Here is a smarter method, based on a data structure called *history graph*.

Definition 7.3. *For $1 \leqslant s \leqslant n$, the* **history graph** *$\mathcal{H}_s$ is a directed acyclic graph whose nodes are all triangles ever been created in the first s rounds. Whenever the algorithm splits a triangle Δ, we add a directed edge from Δ to the three new triangles (Figure 7.3a). Whenever the algorithm flips triangles Δ_1, Δ_2 to Δ'_1, Δ'_2, we add directed edges $\Delta_i \to \Delta'_j$ for $i, j \in \{1, 2\}$. (Figure 7.3b)*

The history graph $\mathcal{H}_s$ contains triangles of outdegrees 3, 2 and 0, where the ones with zero outdegree are exactly the triangles of $\mathcal{D}_s$. It can be built during the incremental construction at asymptotically no extra cost; but it may need extra space to keeps all triangles ever created.

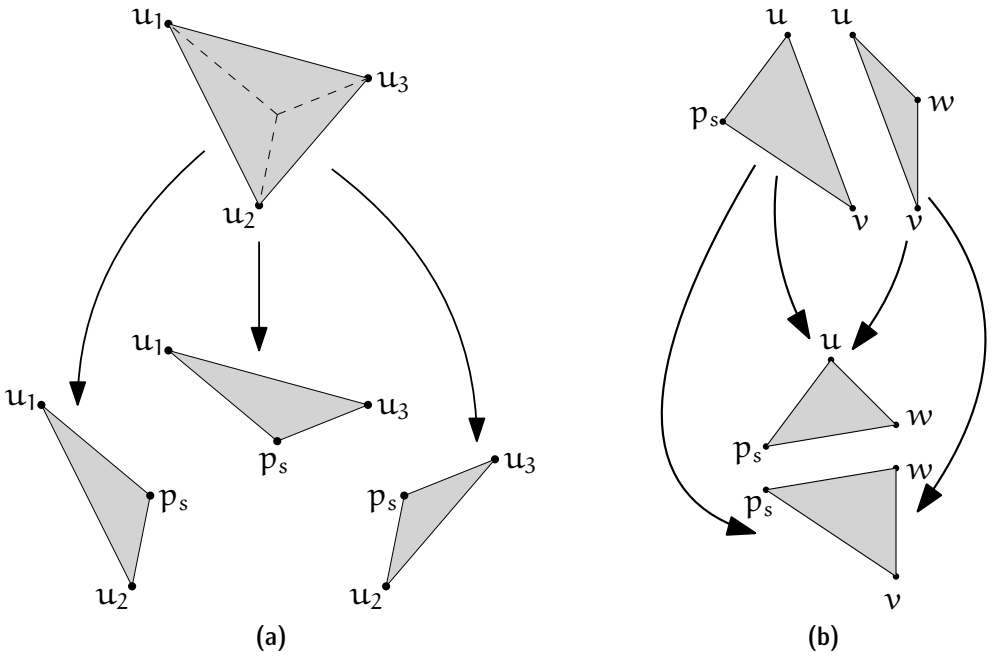

Figure 7.3: *The history graph*

Given $\mathcal{H}_{s-1}$, we can search for the triangle $\Delta \in \mathcal{D}_{s-1}$ that contains p_s by starting from the big triangle abc—it certainly contains p_s—and tracking down a directed path in $\mathcal{H}_{s-1}$. If the current triangle still has outneighbors, we move on to the unique outneighbor containing p_s (recall that we assume general position) and search iteratively. If the current triangle has no outneighbors, it must be in $\mathcal{D}_{s-1}$ and contains p_s, so we are done.

7.4 Analysis of the algorithm

The runtime analysis heavily exploits conditional expectations. Here is a quick refresher. Let X, Y be two random variables in a finite probability space. When we "condition on" variable X, what we mean is to "freeze" or "reveal" the outcome of X as a concrete value. Consequently some randomness dissipates, and the distribution of Y is thus biased. In general this distribution shall depend on the concrete X-value. For example, suppose we sample a uniform permutation π of $\{1, 2, 3\}$, and define $X = \pi(1)$ and $Y = \pi(2)$. So Y by itself is uniformly distributed over $\{1, 2, 3\}$. Conditioning on X shall make Y uniformly distributed on $\{1, 2, 3\} \setminus X$ instead.

The conditional expectation $\mathbb{E}(Y \mid X)$ is defined as the expectation of Y taken with respect to this now-biased distribution. Hence $\mathbb{E}(Y \mid X)$ is a function of X in general. In

our illustrative example,

$$\mathbb{E}(Y \mid X) = \begin{Bmatrix} 2.5, & X = 1 \\ 2, & X = 2 \\ 1.5, & X = 3 \end{Bmatrix} = 3 - \frac{X}{2}.$$

It is easy to prove the so-called total expectation formula $\mathbb{E}(Y) = \mathbb{E}[\mathbb{E}(Y \mid X)]$, but it might be more important to remember its interpretation. To compute $\mathbb{E}(Y)$, we first partition the universe depending on the outcome of X. Then for each part, we compute the expectation $\mathbb{E}(Y \mid X)$ individually, which are our "partial results". Finally, we put these pieces together by a weighted average. One can view this as a natural generalization of elementary counting principle: To count the number of certain objects, we could partition them into several types, count each type individually, and then sum them up.

Cost of Lawson flips. Recall from Corollary 7.2 that $d_s := \deg_{\mathcal{D}_s}(p_s)$ captures the running time of Lawson flips as well as the growth of history graph in round s. This leads us to study the expected value of d_s.

Lemma 7.4. $\mathbb{E}[d_s] \leqslant 6$ *for all* s.

Proof. Let us condition on the set P_s, i.e. we freeze the set of the first s points. Note that the exact ordering of these points is not revealed, and remains uniformly random. In particular, p_s is uniformly distributed in P_s. On the other hand, the Delaunay triangulation $\mathcal{D}_s$ is no longer random because it is uniquely determined by P_s.

Hence $\mathbb{E}[d_s \mid P_s]$ means "the expected degree of p_s in the fixed graph $\mathcal{D}_s = \mathcal{D}_s(P_s)$, where the point p_s is sampled from the fixed set P_s uniformly at random".

Since $\mathcal{D}_s$ is a triangulation on $s+3$ points with triangular convex hull, it follows from Lemma 6.4 that it has $3(s+3)-6$ edges. Excluding the three edges of the convex hull, the total degree of all points in P_s is at most $2(3(s+3)-9) = 6s$. This implies that $\mathbb{E}[d_s \mid P_s] \leqslant 6$. The lemma follows by removing the condition via total expectation. □

By combining the above lemmas, we can also prove the following bound on the expected number of triangles created by the algorithm. Note that this is at the same time a bound on the expected size of the history graph.

Corollary 7.5. *The expected number of triangles ever created in* n *rounds is at most* $9n + 1 = O(n)$. *All the same, the expected running time of all Lawson flips in* n *rounds is* $O(n)$.

Proof. Before inserting any points from the set P, we only have the artificial triangle abc. During round s of the algorithm, we know from Corollary 7.2 that the number of new triangles created is $2d_s - 3$. Combined with Lemma 7.4, the expected number of created triangles in all n iterations is

$$1 + \mathbb{E}\left[\sum_{s=1}^{n} 2d_s - 3\right] = 1 + \sum_{s=1}^{n} (2\mathbb{E}[d_s] - 3) \leqslant 1 + (2 \cdot 6 - 3)n = 9n + 1. \qquad \square$$

Note that we cannot say that every round creates at most 9 triangles; as there could be very costly insertions with some probability. But the claim holds in expectation which is enough to provide a linear expected runtime.

Cost of locating points. We proceed now to the most difficult part of the analysis: to bound the time for finding the triangle in $\mathcal{H}_{s-1}$ that contains p_s. This is proportional to the number of triangles in $\mathcal{H}_{s-1}$ that contains p_s. Hence let us take a closer look at all the triangles in the history graph $\mathcal{H}_{s-1}$.

Suppose a triangle Δ was added to the graph in round r. If $\Delta \in \mathcal{D}_r$ then we call it *valid*, as it survived the round that it was born. Otherwise we call it *ephemeral*, as it got flipped away in the very same round it was born. To make the analysis possible, we want to express the running time in terms of valid triangles only.

Observation 7.6. *The number of triangles in $\mathcal{H}_{s-1}$ that contains p_s is proportional to the number of* valid *triangles in $\mathcal{H}_{s-1}$ whose* circumcircle *contains p_s.*

Indeed, recall from Lemma 7.1 that at every Lawson flip in some round r, one of the replaced triangles is in $\mathcal{D}_{r-1}$ (hence valid) and the other one was created in the current round r (hence ephemeral). That is, a flip always destroys valid and ephemeral triangles in pair. Therefore, for any ephemeral triangle $\Delta \in \mathcal{H}_{s-1}$ that contains p_s, we may charge it to its partner Δ', the valid triangle that was destroyed together with Δ. It is clear that the triangle Δ' is charged at most once. We also know from the condition of Lawson flip that Δ, hence also p_s, is contained in the circumcircle of Δ'. So the observation is established.

Back to time analysis, let us introduce some handy random variables. For every $1 \leqslant r < s \leqslant n$,

- $\tau_r = \mathcal{D}_r \setminus \mathcal{D}_{r-1}$ consists of all triangles in $\mathcal{D}_r$ newly created in round r;
- $\varphi_{r,s}$ is the number of triangles in τ_r whose circumcircle contains the point p_s.

Then the observation implies that the searching time in round s is proportional to $\sum_{r=1}^{s-1} \varphi_{r,s}$. This works since any valid triangle in $\mathcal{H}_{s-1}$ that contains p_s must be in τ_r for some $r < s$.

Instead of bounding this cost for a particular round s, we try to bound the combined cost over all rounds, i.e.

$$T := \sum_{s=1}^{n} \sum_{r=1}^{s-1} \varphi_{r,s} = \sum_{r=1}^{n} \sum_{s=r+1}^{n} \varphi_{r,s}$$

where we exchanged the summations in the second equality.

Lemma 7.7. *It holds that $\mathbb{E}[T] = O(n \log n)$.*

Proof. Using linearity of expectation, we have

$$\mathbb{E}[T] = \sum_{r=1}^{n} \sum_{s=r+1}^{n} \mathbb{E}[\varphi_{r,s}].$$

Observe that the variables $\varphi_{r,r+1}, \varphi_{r,r+2}, \ldots, \varphi_{r,n}$ are identically distributed due to symmetry. To see this more clearly, recall that $\varphi_{r,s}$ is defined in terms of τ_r and p_s. Let us condition on (i.e. freeze) the ordering $(p_1, \ldots, p_r)$. Then τ_r is fixed, whereas each of $p_{r+1}, p_{r+2}, \ldots, p_n$ is uniformly distributed over the fixed set $P \setminus \{p_1, \ldots, p_r\}$. It follows that the variables of interest are identically distributed under the condition; but we may remove the condition nonetheless via total probability.

Hence we may simplify the expectation as

$$\mathbb{E}[T] = \sum_{r=1}^{n} (n - r) \cdot \mathbb{E}[\varphi_{r,r+1}] \tag{7.8}$$

It remains to analyze the expected value $\mathbb{E}[\varphi_{r,r+1}]$ for every particular $1 \leqslant r \leqslant n$. Let Γ consist of all triangles in $\mathcal{D}_r$ whose circumcircle contains p_{r+1}. This is nothing but "all triangles in $\mathcal{D}_r$ that are destroyed in round $r+1$". From Lemma 7.1 we immediately have $|\Gamma| = d_{r+1} - 2$ (we also count the triangle that is split into three).

On the other hand, by definition we may rewrite $\varphi_{r,r+1} = \sum_{\Delta \in \Gamma} X_\Delta$. Here X_Δ is the indicator variable for the event $\Delta \notin \mathcal{D}_{r-1}$, which takes value 1 if the event happens and value 0 otherwise. In order to apply linearity of expectation, the summation must be "derandomized"; that is, it should not run over a random set Γ. Hence we condition on (i.e. freeze) the *set* P_r as well as the point p_{r+1}. We stress that the concrete ordering of P_r is not revealed. Nevertheless, the Delaunay triangulation $\mathcal{D}_r$ is uniquely determined, so is Γ. Therefore,

$$\begin{aligned}
\mathbb{E}[\varphi_{r,r+1} \mid P_r, p_{r+1}] &= \sum_{\Delta \in \Gamma} \mathbb{E}[X_\Delta \mid P_r, p_{r+1}] \\
&= \sum_{\Delta \in \Gamma} \Pr[\Delta \notin \mathcal{D}_{r-1} \mid P_r, p_{r+1}] \\
&\leqslant \sum_{\Delta \in \Gamma} \frac{3}{r} \\
&= \frac{3}{r} \cdot |\Gamma| \;=\; \frac{3}{r} \cdot (d_{r+1} - 2)
\end{aligned}$$

To see the inequality, observe that if a triangle $\Delta \in \Gamma$ is not contained in $\mathcal{D}_{r-1}$, then it must be created in round r. In particular, p_r needs to be its vertex by Lemma 7.1. As p_r is uniformly distributed over the set P_r (under conditions P_r, p_{r+1}), the event happens with probability at most $3/r$. ("At most" because some vertex of Δ might be the artificial points a, b or c; in that case p_r cannot hit it).

Now we remove the condition via total expectation and obtain

$$\mathbb{E}[\varphi_{r,r+1}] \leqslant \frac{3}{r} \cdot (\mathbb{E}[d_{r+1}] - 2) \leqslant \frac{12}{r}, \tag{7.9}$$

where we used Lemma 7.4 in the last step. We are finally able to plug (7.9) back into (7.8) to conclude the proof:

$$\mathbb{E}[T] \leqslant \sum_{r=1}^{n} \frac{12(n-r)}{r} \leqslant 12n \sum_{r=1}^{n} \frac{1}{r} = O(n \log n). \qquad \square$$

The main theorem. Having the previous lemmas at hand, assembling our main result is now straightforward.

Theorem 7.10. *The Delaunay triangulation of a set P of n points in the plane can be computed in $O(n \log n)$ expected time, using $O(n)$ expected space.*

Proof. The correctness of the algorithm follows from the correctness of the Lawson flip algorithm, and from the fact that we perform all possible Lawson flips in each round. For the space consumption, only the history graph might use more than linear space, but Lemma 7.5 bounds its expected size by $O(n)$, so the claim follows.

For the running time, Lemma 7.7 bounds the expected time spent on point location (over all n rounds) by $O(n \log n)$, and Lemma 7.5 bounds the expected time spent on Lawson flips (over all n rounds) by $O(n)$. So the algorithm runs in $O(n \log n)$ time in expectation. $\square$

Exercise 7.11. *For a sequence of n pairwise distinct numbers $y_1, \ldots, y_n$ consider the sequence of pairs $(\min\{y_1, \ldots, y_i\}, \max\{y_1, \ldots, y_i\})_{i=0,1,\ldots,n}$ ($\min \emptyset := +\infty, \max \emptyset := -\infty$). How often do these pairs change in expectation if the sequence is permuted randomly, each permutation appearing with the same probability? Determine the expected value.*

Exercise 7.12. *Given a set P of n points in convex position represented by the clockwise sequence of the vertices of its convex hull, provide an algorithm to compute its Delaunay triangulation in $O(n)$ time.*

Questions

31. *What conditions should the three "far-out" points a, b, c satisfy? Explain the reason.*
32. *Describe the algorithm for the incremental construction of $\mathcal{DT}(P)$*: how do we find the triangle containing the point p_s to be inserted into $\mathcal{D}_{s-1}$? How do we transform $\mathcal{D}_{s-1}$ into $\mathcal{D}_s$? How many steps does the latter transformation take?
33. *What are the two types of triangles that the history graph contains?*