

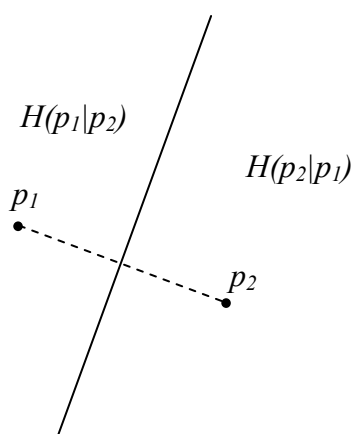
Лекции 10-11. Разбиения Вороного и триангуляции Делоне

1. Разбиение Вороного

Рассмотрим следующую задачу.

На плоскости задано множество S , состоящее из N точек. Требуется для каждой точки p_i из S определить на плоскости геометрическое место точек (так называемый *локус*), для которых расстояние до точки p_i не больше, чем до любой другой точки множества S . Такое множество точек называется ячейкой Вороного полюса p_i . Другие названия: полигон Вороного, многоугольник Вороного (Георгий Феодосьевич Вороной (1868-1908) – русский математик, исследовавший теорию квадратичных форм применительно к задачам кристаллографии).

Если $S = \{p_1, p_2\}$ состоит из двух точек, то ячейки Вороного полюсов p_1 и p_2 являются полуплоскостями, ограниченные прямой – серединным перпендикуляром отрезка $[p_1, p_2]$ (так называемым *бисектором*).

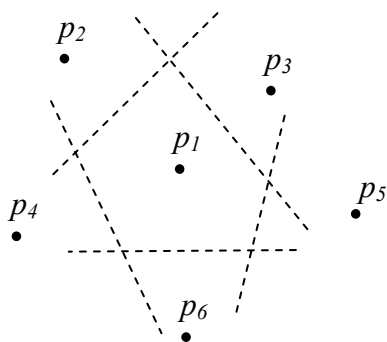


Обозначим множество точек, расстояние от которых до p_i не больше, чем до p_j , через $H(p_i|p_j)$. Очевидно, что это полуплоскость. Тогда ячейка Вороного $V(p_i)$ полюса p_i представляет собой пересечение таких полуплоскостей $V(p_i) = \bigcap_{i \neq j} H(p_i | p_j)$.

Очевидно, что ячейка Вороного представляет собой часть плоскости, ограниченную прямыми линиями, либо лучами, либо отрезками. Отсюда и название полигон Вороного. Заметим, что такой полигон может лежать в конечной части плоскости, либо

уходить в бесконечность.

Если построить полигоны Вороного для всех точек из множества S , то они покроют всю плоскость. Такое представление плоскости в виде объединения полигонов Вороного называется *разбиением Вороного*. Совокупность границ полигонов называется *диаграммой Вороного*. Диаграмма Вороного состоит из лучей и отрезков прямых, называемых ребрами. Точки пересечения ребер называются вершинами.



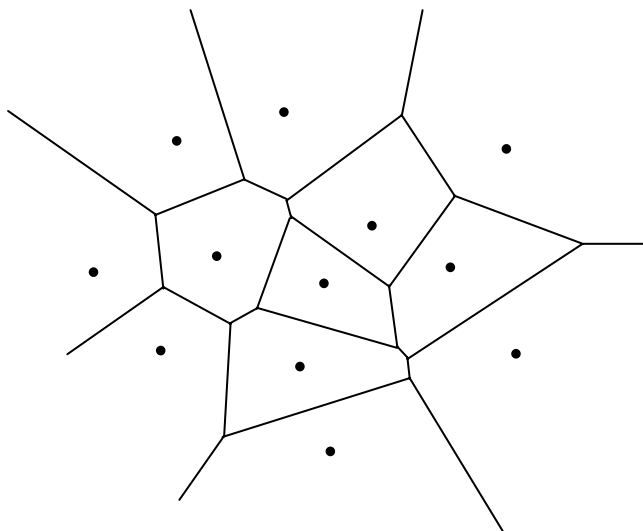
Предположим, что точки в множестве S являются точками общего положения, т.е. никакие три из них не лежат на одной прямой и никакие четыре – на одной окружности.

Тогда справедливы следующие утверждения относительно свойств диаграммы Вороного.

1. Каждая вершина диаграммы Вороного является точкой пересечения в точности трех ребер.
2. Каждая вершина диаграммы Вороного является центром описанной окружности трех точек-полюсов из S , причем внутри этой окружности не содержится точек из S .

3. Полигон Вороного полюса p_i является неограниченным тогда и только тогда, когда p_i лежит на границе выпуклой оболочки S .

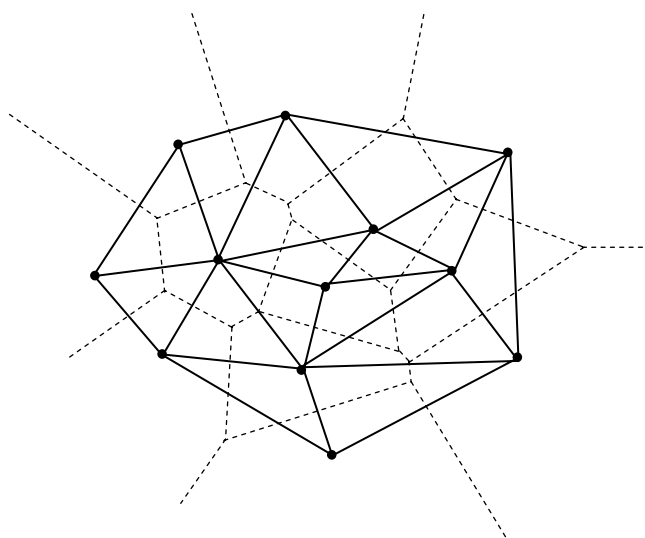
Будем считать, что все лучи диаграммы Вороного, уходящие в бесконечность, «соединяются» в некоторой бесконечно удаленной точке, также являющейся полюсом. Тогда диаграмму можно рассматривать как плоский прямолинейный граф.



2. Триангуляция Делоне

Рассмотрим плоский граф, двойственный к диаграмме Вороного. Для этого соединим отрезками прямых линий те точки из S , которые имеют смежные полигоны Вороного. Получаемый при этом граф является триангуляцией, названной именем Бориса Николаевича Делоне, впервые рассмотревшего этот объект в работе, опубликованной в 1934 году.

Справедливы следующие свойства двойственного графа диаграммы Вороного.



1. Он является триангуляцией выпуклой оболочки множества точек S . Это значит, что он является плоским прямолинейным графом, каждая грань его представляет собой треугольник и эти треугольники покрывают многоугольник, являющийся выпуклой оболочкой S .

2. Каждая треугольная грань имеет описанную окружность с центром в вершине диаграммы Вороного, проходящую через три полюса и не содержащую внутри себя других полюсов из S . Задаваемый такой окружностью круг называется *пустым кругом*.

Любой конечный набор точек на плоскости, за исключением некоторых тривиальных случаев, допускает более одного способа триангуляции (речь идет о произвольной триангуляции, а не триангуляции Делоне). Однако при этом сохраняется следующее замечательное свойство.

Теорема о триангуляции набора точек. Пусть множество S содержит $n \geq 3$ точек и не все они лежат на одной прямой и кроме того i точек являются внутренними, т.е. не лежат на границе выпуклой оболочки множества S . Тогда при любом способе триангуляции количество треугольников составит $n+i-2$.

Это свойство, в частности, справедливо и для триангуляции Делоне.

Если S состоит из точек общего положения, то триангуляция Делоне является единственной триангуляцией, у которой описанные окружности граней являются пустыми. Следовательно, это свойство является характеристическим свойством триангуляции Делоне, что позволяет дать следующее определение ее, не связанное с разбиением Вороного.

Триангуляция выпуклой оболочки конечного множества точек, у которой все описанные окружности треугольных граней являются пустыми, называется триангуляцией Делоне.

Это определение позволяет разрабатывать алгоритмы построения триангуляции Делоне, не связанные с построением разбиения Вороного.

Прежде чем рассматривать алгоритмы построения триангуляции Делоне, покажем, как некоторые важные задачи вычислительной геометрии могут быть решены с помощью этого объекта.

3. Близость точек

Пусть S – конечное множество точек на плоскости.

Задача 1 (ближайшая пара). Найти пару точек в S , расстояние между которыми минимально.

Задача 2 (ближайшие соседи). Для каждой точки из S найти ближайшую точку среди остальных точек из S .

Предположим, что триангуляция Делоне для точек S построена. Тогда решения этих задач могут быть получены из следующего очевидного утверждения: каждая точка в S соединяется со своей ближайшей точкой ребром в триангуляции Делоне множества точек S .

На этом основании для решения задачи 1 достаточно найти самое короткое ребро в триангуляции. Концевые точки этого ребра и будут ближайшей парой. А для решения задачи 2 нужно для каждой точки найти самое короткое ребро среди всех инцидентных ей ребер. Это кратчайшее ребро и определит ближайшего соседа рассматриваемой точки.

Оценим вычислительную сложность этого решения. На основании теоремы Эйлера известно, что количества вершин v , ребер e и внутренних граней f в плоском графе связаны формулой $e = v + f - 1$. Поскольку для триангуляции, у которой имеется i внутренних вершин, $f = v + i - 2$, имеем $e = 2v + i - 3$. Но $i < v$, поэтому справедлива оценка $e < 3v$ или в асимптотической нотации $e = O(v)$.

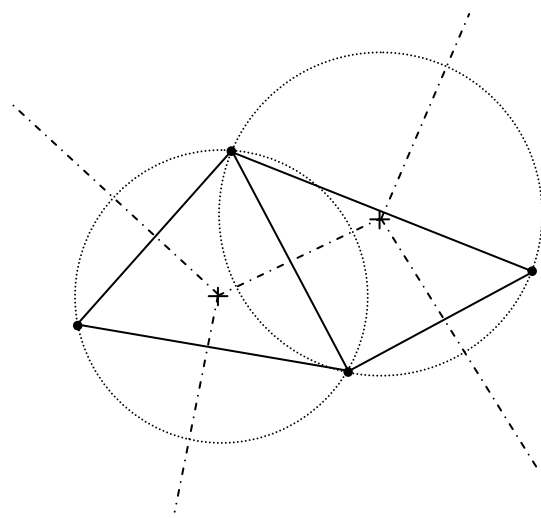
При решении задачи 1 нужен однократный просмотр всех ребер триангуляции, а при решении задачи 2 каждое ребро рассматривается ровно два раза (при переборе инцидентных ребер для одной концевой вершины, а потом

Таким образом, мы показали, что задача построения минимального покрывающего графа $N\log N$ -сводима к задаче построения триангуляции Делоне.

5. Преобразования двойственных графов

Из диаграммы Вороного легко строится триангуляция Делоне. Для этого, согласно данному выше определению, достаточно соединить отрезками прямых полюса, имеющие смежные полигоны Вороного. Количество таких смежных пар полюсов равно количеству ребер в триангуляции Делоне, т.е. составляет $O(N)$. Следовательно, для того, чтобы построить триангуляцию Делоне из разбиения Вороного, необходимо время $O(N)$.

Рассмотрим обратную задачу: каким образом получить диаграмму Вороного из триангуляции Делоне. Решение ее очевидно, поскольку вершинами диаграммы Вороного являются центры описанных кругов треугольников-граней триангуляции Делоне, а ребрами – серединные перпендикуляры ребер триангуляции.



Количество вершин диаграммы Вороного равно количеству граней триангуляции Делоне, а количества ребер у них одинаковы. Таким образом, получается, что построение диаграммы Вороного на основе триангуляции Делоне может быть выполнено за время $O(N)$.

6. Простой алгоритм построения триангуляции Делоне

Мы рассмотрим несколько идей построения триангуляции Делоне и начнем с простейшего – «наивного» алгоритма. Для всех рассматриваемых алгоритмов входом является множество S из N точек.

Алгоритм отбора треугольных граней

- 1) Перебор всех возможных треугольников, т.е. троек точек из S . Это требует времени $O(N^3)$.
- 2) Для каждого треугольника построить описанную окружность (требуемое время $O(1)$) и проверить, пуста ли она. Такая проверка сводится к последовательной проверке попадания $N-3$ полюсов внутрь окружности (время $O(N)$). Те треугольники, чьи окружности являются пустыми, объявляются гранями триангуляции Делоне и запоминаются.
- 3) Определить смежность треугольных граней на основе перебора всех пар треугольников. Поскольку количество треугольных граней $O(N)$, этот шаг потребует $O(N^2)$ времени.

Как нетрудно понять, этот алгоритм требует времени $O(N^4)$. Практическая реализация этого алгоритма не составит труда, однако его эффективность представляется слишком низкой.

7. Алгоритм жадной триангуляции

Жадными называются алгоритмы, которые в процессе построения некоторого объекта ничего не создают лишнего. В данном случае имеется в виду, что все образуемые треугольные грани являются окончательными и не корректируются и не уничтожаются в процессе работы алгоритма.

Алгоритм работает путем наращивания текущей триангуляции по одному треугольнику за один шаг. Вначале текущая триангуляция состоит из единственного ребра оболочки, по окончании работы алгоритма текущая триангуляция становится триангуляцией Делоне. На каждой итерации алгоритм ищет новый треугольник, который подключается к границе текущей триангуляции.

Введем следующую классификацию ребер триангуляции Делоне относительно текущей конфигурации:

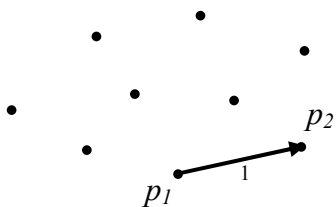
- открытое ребро – такое, у которого обнаружена лишь одна примыкающая к нему грань;
- закрытое ребро – такое, у которого обнаружены обе примыкающие грани.

При этом для ребра, входящего в границу выпуклой оболочки множества S , считается, что одной из примыкающих граней является неограниченная часть плоскости, лежащая вне выпуклой оболочки, которую будем называть «внешней гранью».

Алгоритм последовательного присоединения граней

Шаг 1. Найти первое ребро триангуляции, являющееся ребром выпуклой оболочки S . Это можно сделать, например, следующим образом.

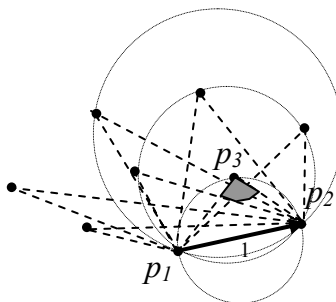
- найти точку p_1 с минимальной ординатой;
- найти точку p_2 такую, чтобы вектор p_1p_2 составлял минимальный угол с осью абсцисс.



Полученное ребро (p_1, p_2) является ребром триангуляции. Оно объявляется открытым ребром, поскольку справа к нему примыкает внешняя грань, а слева пока не примыкает ни одной грани. Это ребро помещается в *список открытых ребер*. В приведенном на рисунке примере найденное ребро обозначено вектором

для того, чтобы было видно, с какой стороны оно является открытым – слева. Номер на ребре показывает его расположение в списке открытых ребер.

Шаг 2. Это основная итерация алгоритма. На каждой итерации выбирается очередное открытое ребро из списка и подвергается обработке, заключающейся в построении либо поиске примыкающей к нему грани. Условием завершения работы алгоритма является исчерпание открытых ребер.

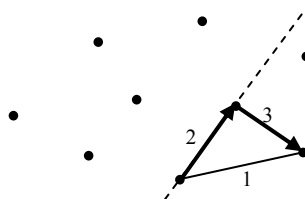


Рассмотрим более подробно, каким образом обрабатывается открытое ребро. Возникающие при этом случае удобно анализировать на приведенном примере.

Для открытого ребра 1 мы хотим найти точку из S , дополняющую его до треугольной грани. Эта точка p_3 должна выбираться из следующих соображений. Во-первых, она должна лежать слева от p_1p_2 . А во-вторых,

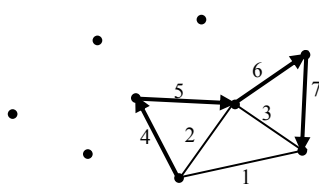
описанная окружность для полученного треугольника $p_1p_2p_3$ должна быть была пустой. Как видно из рисунка, это достигается в том случае, когда угол $p_1p_2p_3$

является максимальным среди всех возможных. Этот угол принято называть «угол, под которым ребро p_1p_2 видно из точки p_3 ». Таким образом, правило выбора третьей вершины для образуемой треугольной грани состоит в следующем. Нужно выбрать точку, лежащую слева от ребра p_1p_2 , из которой это ребро видно под максимальным углом. Поскольку все такие углы лежат в диапазоне $[0, \pi]$, выбор максимального угла соответствует выбору угла с минимальным косинусом. А для расчета косинуса можно воспользоваться следующей формулой, выражающей его через скалярное произведение векторов: $\cos(\angle p_1p_3p_2) = \frac{(p_3p_1, p_3p_2)}{|p_3p_1| \cdot |p_3p_2|}$.



Таким образом, нами найдена новая грань – треугольник $p_1p_2p_3$, примыкающая к открытому ребру p_1p_2 . Это ребро объявляется закрытым и создаются два новых открытых ребра (2 и 3), которые помещаются в список открытых ребер.

На следующей итерации обработке подвергается очередное открытое ребро 2. На этом шаге нет каких-то дополнительных особенностей по сравнению с предыдущей итерацией. Отметим лишь, что в качестве кандидатов для построения грани рассматриваются лишь точки, лежащие слева от этого ребра (на рисунке они

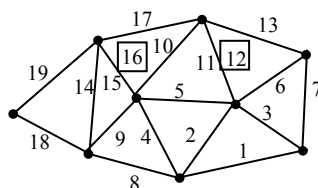


отделены пунктирной линией). В результате ребро 2 становится закрытым и образуются открытые ребра 4 и 5. Аналогично обрабатывается и следующее открытое ребро 3. Оно становится после обработки закрытым и при этом образуются ребра 6 и 7.

Обработка ребер 4 и 5 также не имеет каких-то особенностей, а вот для ребра 6 возникает ситуация, которой до этого не было. Здесь образуются два новых открытых ребра: 12 и 13. При этом, как видно из рисунка, ребро 12 просто совпадает с ребром 11, но имеет с ним противоположные направления. Это означает, что ребро 12 не может быть объявлено открытым, поскольку примыкающей гранью к нему оказывается треугольник, образованный ранее ребрами 5, 10 и 11.

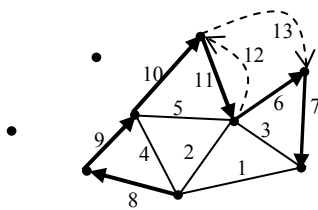
Возникающая здесь ситуация подсказывает, что на шаге 2 основного алгоритма следует осуществить проверку вновь создаваемых открытых ребер на совпадение с уже существующими открытыми ребрами. Если такое совпадение установлено, то выполняется «склейка» этих ребер. В нашем примере склейка состоит в следующем. Ребро 12 не создается, а ребро 11 становится закрытым и к нему примыкает грань из ребер 6, 11 и 13.

Еще одна особенность проявляется при обработке ребра 7. Здесь оказывается, что не существует точек из S , лежащих левее ребра. Это означает, что ребро принадлежит выпуклой оболочке S и к нему примыкает внешняя грань. Такое ребро отмечается как закрытое без построения треугольной грани.



На рисунке показан окончательный вид триангуляции с нумерацией ребер в последовательности их построения и обработки.

Ребра, для которых была выполнена склейка (12 и 16), обведены квадратиками.



Таким образом, на каждой итерации шага 2 выполняется один просмотр всех точек из S с целью проверки, лежат ли они левее обрабатываемого ребра, и вычисления косинуса угла визирования. Затем, возможно, образуется пара новых ребер, для которых проверяется появление склеек с открытыми ребрами из списка.

Количество открытых ребер в списке составляет $O(N)$. Это означает, что число итераций есть $O(N)$, и время выполнения одной итерации есть $O(N)$. А это дает общую оценку временной сложности рассматриваемого алгоритма жадной триангуляции $O(N^2)$.

8. Триангуляция методом «разделяй и властвуй»

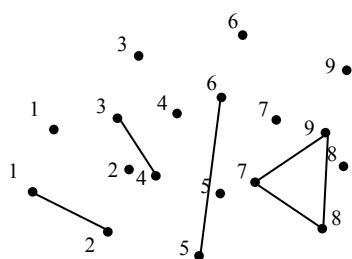
Для того, чтобы оценить минимально необходимое время для построения триангуляции Делоне, рассмотрим одномерный случай – когда все N точек из S расположены на одной прямой. Если некоторый алгоритм строит триангуляцию такого множества точек, то, в частности, эта триангуляция упорядочивает точки вдоль прямой, на которой они расположены. Это значит, что с помощью такого алгоритма можно решить задачу сортировки массива из N чисел. Поскольку нижняя оценка для задачи сортировки известна и равна $\Omega(N \log N)$, то получается, что и для задачи построения триангуляции Делоне справедлива та же самая нижняя оценка.

Алгоритм, описанный ниже, имеет верхнюю оценку временной сложности $O(N \log N)$ и, следовательно, является оптимальным алгоритмом.

Идея этого алгоритма состоит в использовании парадигмы «разделяй и властвуй»:

1. Разделить множество S на два приблизительно равных подмножества S_1 и S_2 , используя для этого медиану по координате x .
2. Рекурсивно построить триангуляции Делоне для множеств S_1 и S_2 .
3. Слить триангуляции, т.е. из триангуляций S_1 и S_2 построить триангуляцию S .

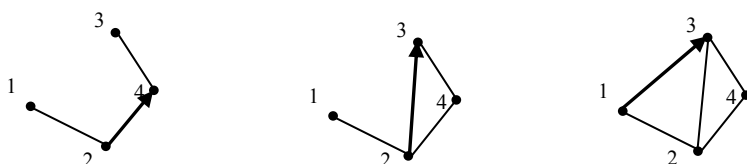
Для выполнения первого шага этого подхода, в том числе и при рекурсивных построениях триангуляций подмножеств, нужно произвести лексикографическое упорядочение точек из S по паре координат x и y . Такое упорядочение требует времени $O(N \log N)$. Как было показано ранее, если удастся предложить процедуру слияния рекурсивно построенных триангуляций множеств S_1 и S_2 , имеющую временную сложность $O(N)$, то общий алгоритм и будет иметь сложность $O(N \log N)$. Таким образом, главным предметом внимания должен стать алгоритм слияния двух разделенных триангуляций Делоне. Разделенными они являются потому, что выпуклые оболочки множеств S_1 и S_2 не пересекаются.



В примере на рисунке множество S состоит из девяти точек $S = \{1, \dots, 9\}$. Точки пронумерованы в соответствии с лексикографическим упорядочением. После деления образуются два подмножества $S_1 = \{1, \dots, 4\}$ и $S_2 = \{5, \dots, 9\}$. В свою очередь, при рекурсивном построении триангуляции множества S_1 оно разбивается на два подмножества $S_{11} = \{1, 2\}$ и $S_{12} = \{3, 4\}$. Каждое из этих подмножеств состоит из двух точек.

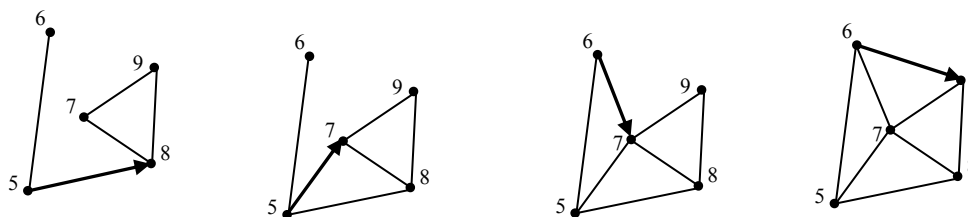
Триангуляция Делоне двухточечного множества имеет тривиальный вид: она содержит одно ребро, соединяющее эти точки. Аналогично при рекурсивном построении триангуляции S_2 , получаются в результате его разделения два подмножества $S_{21}=\{5,6\}$ и $S_{22}=\{7,8,9\}$. Триангуляции этих подмножеств также легко строятся. Для двухточечного множества это единственное ребро, а для трехточечного – одна треугольная грань и три ограничивающих ее ребра.

Из триангуляций S_{11} и S_{12} построим методом слияния триангуляцию S_1 . Такое слияние начинается с построения нижней касательной к обоим множествам S_{11} и S_{12} . Нижней касательной называется вектор, началом которого является точка из левого подмножества, а концом – точка из правого подмножества, относительно которого все остальные точки лежат слева. Такой нижней касательной является вектор с началом в точке 2 и концом в точке 4. Очевидно, что ребро (2,4) входит в выпуклую оболочку S_{11} и в триангуляцию Делоне этого множества. Теперь к этому ребру нужно присоединить



треугольную грань. Кандидатами для третьей вершины являются точки 1 и 3. Выбрать нужно ту из них, из которой ребро (2,4) видно под максимальным углом. Такой вершиной является точка 3. Образует треугольную грань (2,3,4) и ребро (2,4) объявляем открытым ребром. Следующим шагом строится новая грань, примыкающая к этому ребру – треугольник (1,3,2) и триангуляция множества S_1 готова.

Триангуляция множества S_2 строится путем слияния триангуляций S_{21} и S_{22} аналогичным образом. В качестве некоторой особенности следует отметить, что при обработке открытого ребра (5,7) в качестве кандидата рассматривается лишь точка 6, поскольку лишь она лежит слева от ребра.

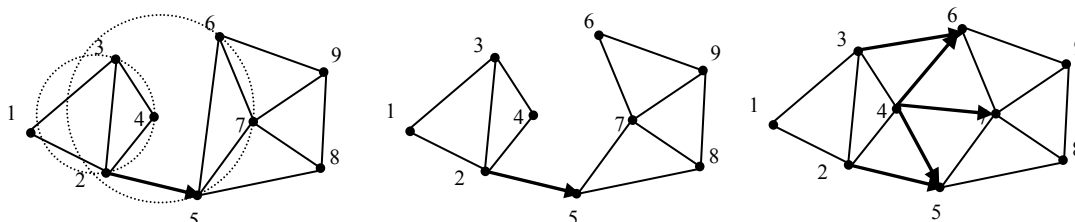


Рассмотренные шаги алгоритма были достаточно простыми, поскольку они описывают слияние тривиальных триангуляций: двухточечных и трехточечных множеств.

Следующий шаг состоит в слиянии триангуляций S_1 и S_2 . Он уже демонстрирует достаточно общую ситуацию, возникающую при слиянии нетривиальных триангуляций. Процесс слияния вновь начинается с построения нижней касательной к непересекающимся выпуклым оболочкам S_1 и S_2 . В лекции 7 описан такой алгоритм, имеющий вычислительную сложность $O(N)$. В примере такой нижней касательной является вектор (2,5). Однако здесь, в отличие от предыдущих более простых случаев, нельзя немедленно строить новую грань, присоединяя к ребру (2,5) вершину 4 или 6. Это связано с тем, что треугольные грани (2,3,4) и (5,6,7), которые имели пустые описанные круги в своих триангуляциях S_1 и S_2 соответственно, могут утратить это свойство относительно объединенного множества точек S . Действительно, в пустой круг

треугольника (5,6,7) попадает точка 2, поэтому он не войдет в строящуюся триангуляцию. Этот треугольник должен быть разрушен.

После этого следует проверить, не попадают ли точки 2 и 4 в описанный круг треугольника (5,7,8) и точки 5 и 7 в описанный круг треугольника (2,3,4). Поскольку этого не происходит, строим новую грань, рассматривая в качестве кандидатов точки 4 и 7. Весь процесс последовательного построения граней показан на рисунке.



Общий алгоритм слияния двух триангуляций, таким образом, описывается следующими шагами.

1. Найти нижнюю касательную к обеим триангуляциям и объявить ее открытым ребром.
2. Рассмотреть смежные ребра для открытого ребра: против часовой стрелки на левой триангуляции и по часовой стрелке на правой. Если эти ребра являются граничными для треугольных граней, то провести тестирование этих граней. Тестирование грани заключается в проверке попадания в ее описанный круг концевых точек противоположного ребра. В случае, если тестирование дает положительный результат (т.е. круг не пуст), соответствующая грань разрушается, оболочка триангуляции корректируется и нужно повторно выполнить пункт 2, иначе перейти к пункту 3.
3. Построить новую треугольную грань, примыкающую к открытому ребру. В качестве кандидатов рассмотреть только две точки: концы смежных с ребер из оболочек триангуляций. Выбор третьей вершины грани осуществляется по критерию максимального угла, под которым видно открытое ребро из этой вершины. Если новую грань образовать не удалось, это значит, что построение триангуляции закончено. В противном случае образовать новое открытое ребро и перейти к пункту 2.

Оценим вычислительную сложность процесса слияния. Общее число разрушенных треугольников не может превзойти их суммарного количества в обеих сливаемых триангуляциях, т.е. $O(N)$. Образование новой грани требует времени $O(1)$. И как уже отмечалось, поиск нижней касательной требует также $O(N)$ времени. Следовательно, общее время на слияние есть $O(N)$. А отсюда следует, что время решения всей задачи $O(N \log N)$.