Описание алгоритма за $O(n^4)$

Алгоритм принимает множество S из n непересекающихся отрезков. P - множесто концов отрезков S, является МТОП (никакие 3 точки не лежат на одной прямой).

Доказательство основных утверждений алгоритма приведены в приложении.

1. Строится множество L, прямых, определяющихся любой парой $p_i, p_j \in P$.

   Данная операция требует $O(n^2)$ времени, $|L| = 2n^2 - n$.

   Для дальнейшей обработки множество точек P сохраняется. Для каждого отрезка запоминаются индексы его концов.

2. Строится упорядочение прямых $A(L)$ с помощью инкрементального алгоритма. Его сложность для m прямых есть $O(m^2)$. Соответственно в данном случае потребуется $O(n^4)$.

   Результатом работы данного алгоритма является РСДС, занимающий линейную память от числа ребер, вершин и граней. Так как это число для $A(L)$ квадратично зависит от количества прямых в L, то расход по памяти на данном этапе алгоритма равен $O(n^4)$.

   Для каждого ребра сохраняется информация об отрезках из S, на точках которых была построена прямая, частью которой является данное ребро.

   Также каждое ребро помечается булевым флагом, определяющим лежит оно между задающих его точек или с одной из сторон.

3. Произвольно выбирается грань f_0 РСДС. Производится упорядочивание P по возрастанию полярного угла относительно внутренней точки грани. Упорядоченное множество P обозначим через P_{f_0}.

   Введем понятие «правильность»: f - грань РСДС, q - внутренняя точка f. Назовем отрезок правильным относительно f, если его точки являются соседними в P_f, а тройка qp_ip_j образовывает левый поворот, в преположении, что p_i находится в P_f раньше. Если в S всего один отрезок, то он правильный.

   Все отрезки проверяются на правильность. Обозначим через R_{f_0} количество правильных отрезков относительно.

   P_f и R_f вместе образуют статус.

   Статус не зависит от выбора точки внутри f.

   Построение P_f требует $O(nlog(n))$, так как может быть реализованно сортировкой. Проверка на правильность - $O(n)$, т.к. это константная операция, применяемая ко всем отрезкам.

4. Производится обход РСДС, начиная с грани f_0. На каждом шаге, зная перешагиваемое ребро, имеем информацию об индексах i и j, порождающих его точек, информацию о положении ребра относительно этих точек. Также имеем информацию об отрезке/отрезках, частью которого/которых являются точки p_i, p_j.

   В процессе перешагивания ребра статус меняется для ≤ 2 отрезков. Из-за этого имеющейся информации и статуса достаточно, чтобы внести изменения в последний (обозначим грань до перешагивания f_m, после f_k):

(a) Подсчитываем количество правильных отрезков, содержащих p_i, p_j, до перешагивания. Обозначим через r_m.

(b) Меняем или не меняем точки p_i, p_j местами. Тем самым получаем P_{f_k} из P_{f_m}.

(c) Подсчитываем количество правильных отрезков, содержащих p_i, p_j, после перешагивания. Обозначим через r_k.

(d) $R_{f_k} = R_{f_m} - r_m + r_k$.

На внесение изменений в порядок точек в контейнере и количество правильных отрезков требуется константное время. Таким образом, обработка каждого ребра требует $O(1)$ времени и памяти.

Обход продолжается до тех пор пока не найдется грань, в которой $R_f = n$, или пока не останется непосещенных граней. В первом случае ответом алгоритма - «да» с предоставлением любой точки внутри найденной грани, во втором случае ответ - «нет».

Обход требует $O(N^4)$ времени, по количеству граней в РСДС.

Построение множества всех прямых требует $O(n^2)$ времени, дополнительная обработка отрезков $O(n)$. Построение упорядочения прямых требует $O(n^4)$. Дополнительная обработка ребер РСДС также $O(n^4)$. Подсчет статуса для первой грани требует $O(nlog(n))$. Обход требует $O(n^4)$. Общее время работы: $O(n^2)+O(n)+O(n^4)+O(n^4)+(nlog(n))+O(n^4) = O(n^4)$.

Отрезки, точки и дополнительная информация о них занимают $O(n)$ памяти, множество L требует $O(n^2)$, РСДС $O(n^4)$. Общее требование к памяти: $O(n) + O(n^2) + O(n^4) = O(n^4)$.

Источники:

1. de Berg, Mark; Cheong, Otfried; van Kreveld, Marc; Overmars, Mark (2008). Computational Geometry, Algorithms and Applications (3rd ed.). Springer. pp. 172–177. ISBN 978-3-540-77973-5.