Описание случаев поведения алгоритма в процессе «перешагивания» ребра

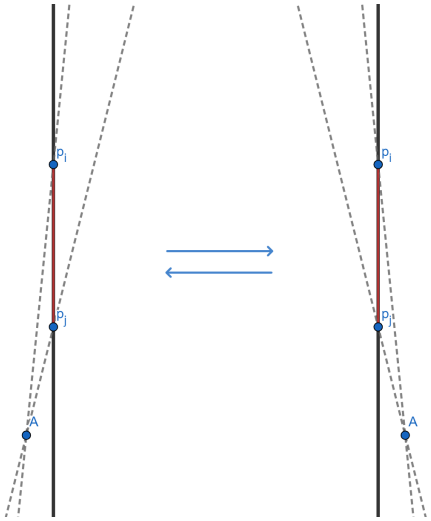
Содержание статуса:

1. Упорядоченные по полярному углу точки отрезков.
2. Информация о правильности всех отрезков.
3. Информация о порядке точек внутри отрезков.

На каждом шаге важным является константный доступ к затрагиваемым отрезкам, информации об их правильности, внутреннем порядке точек.

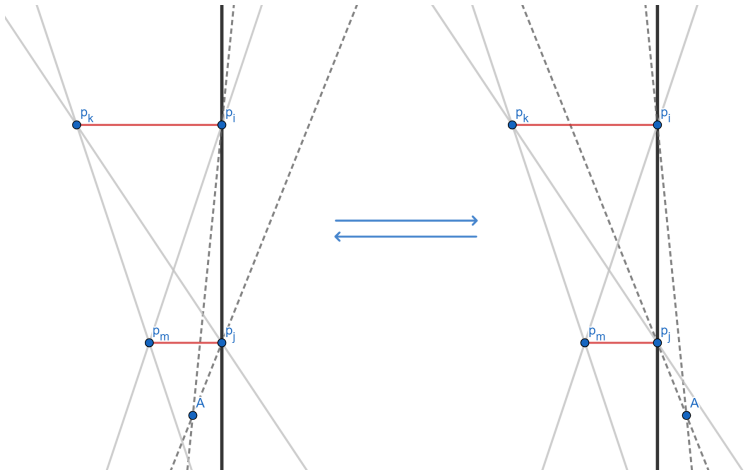
Не умоляя общности, рассмотрим случаи для ребра расположенного вертикально:

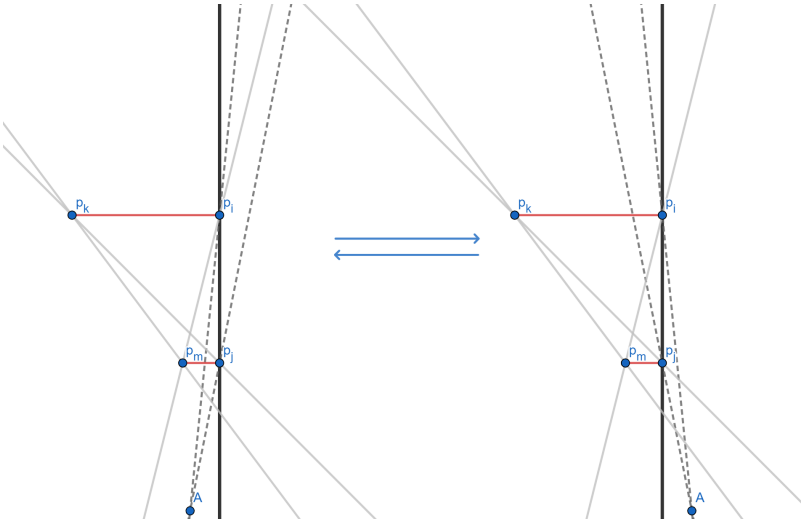
1. Точки p_i, p_j являются точками одного отрезка. Так как перешагивание отрезка исключено из рассмотрения, возможен только вариант представленный ниже, а также зеркальный к нему.

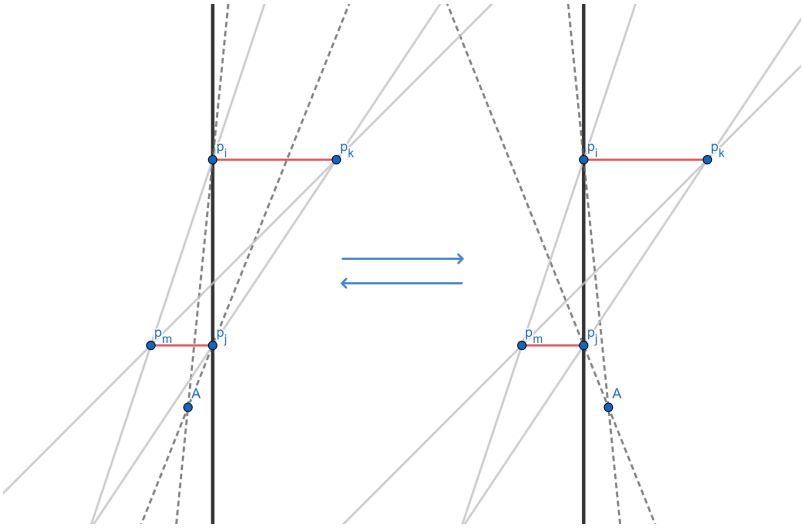


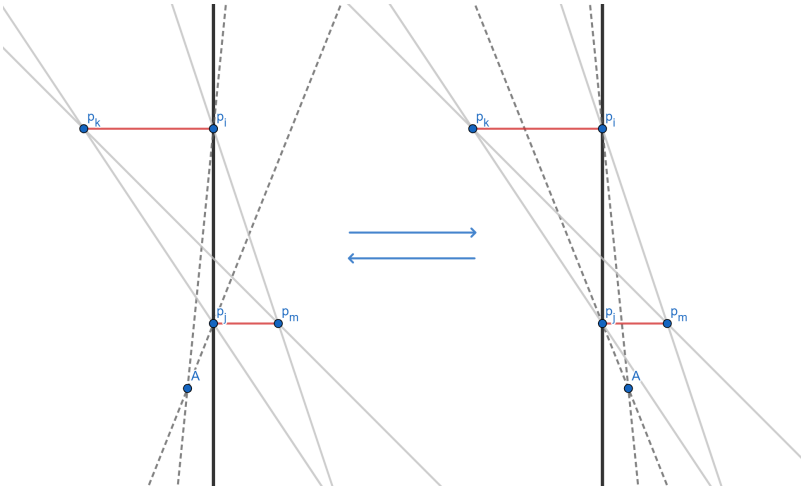
2. Точки p_i, p_j лежат на прямой с одной стороны относительно «перешагиваемого» ребра. Возможны варианты, изображенные ниже, а также зеркальные к ним.

   Выбор какую из точек выбрать за p_i неважен.

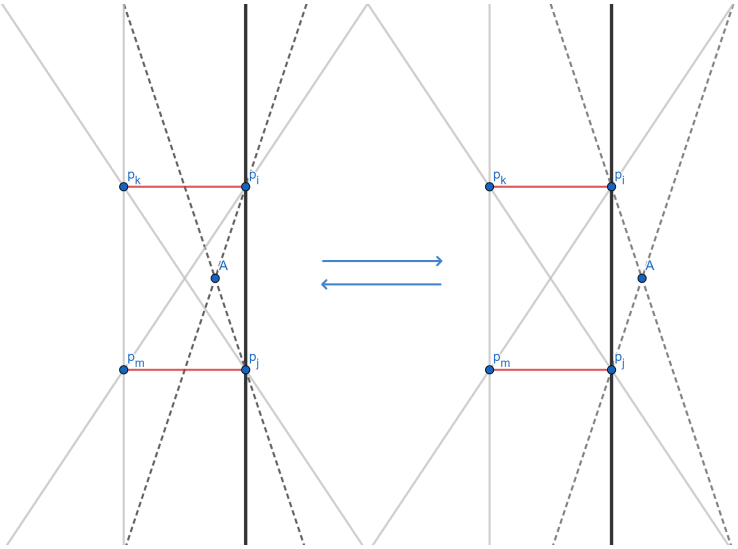


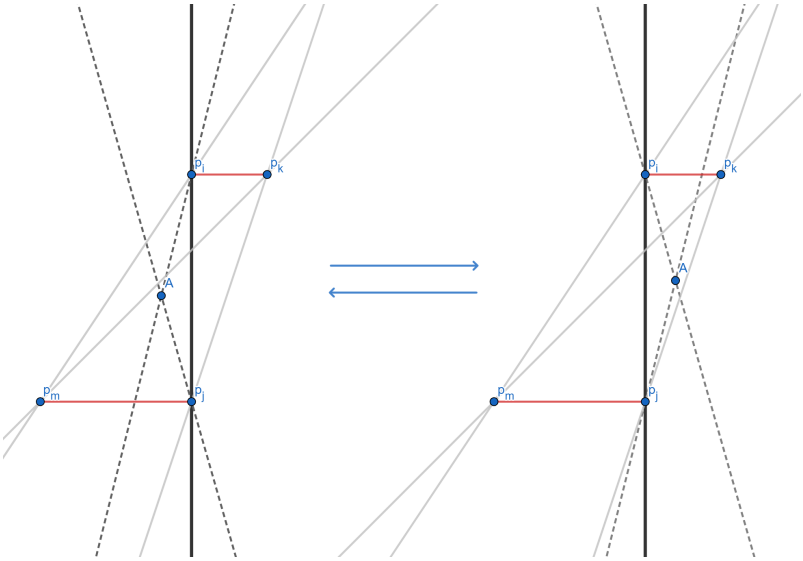






3. Точки p_i, p_j лежат на прямой по разные стороны относительно «Перешагиваемого» ребра. Возможны варианты, изображенные ниже, а также зеркальные к ним.





Таблицы изменений статуса для каждого случая.

Случаи разбиты на пары (по сути "до-после").

Последние 2 пары выделены отдельно. 6 пару можно определить однозначно только если держать в ребре информацию об относительном положении порождающих его точек (с одной стороны/с разных сторон).

Правильность: "+ правильный, "− неправильный, "? если точки рядом, то правильный.

номер	порядок в массиве	правильность	порядок в отрезке
1.1	$..., p_j, p_i, ...$	$+$	p_j, p_i
1.2	$..., p_i, p_j, ...$	$+$	p_i, p_j
2.1	$p_j, ..., p_i, ..., p_k, ..., p_m$	$?, -$	$p_i, p_k : p_j, p_m$
2.2	$p_i, ..., p_j, ..., p_k, ..., p_m$	$-, -$	$p_i, p_k : p_j, p_m$
3.1	$p_j, ..., p_i, ..., p_m, ..., p_k$	$-, -$	$p_i, p_k : p_j, p_m$
3.2	$p_i, ..., p_j, ..., p_m, ..., p_k$	$?, -$	$p_i, p_k : p_j, p_m$
4.1	$p_j, ..., p_i, ..., p_m, ..., p_k$	$-, -$	$p_k, p_i : p_j, p_m$
4.2	$p_i, ..., p_j, ..., p_m, ..., p_k$	$?, ?$	$p_k, p_i : p_j, p_m$
5.1	$p_j, ..., p_i, ..., p_k, ..., p_m$	$?, ?$	$p_i, p_k : p_m, p_j$
5.2	$p_i, ..., p_j, ..., p_k, ..., p_m$	$-, -$	$p_i, p_k : p_m, p_j$

порядок в массиве	правильность	порядок в отрезке	
6.1	$p_j, ..., p_i, ..., p_k, ..., p_m$	$?, ?$	$p_i, p_k : p_m, p_j$
6.2	$p_j, ..., p_i, ..., p_k, ..., p_m$	$?, ?$	$p_i, p_k : p_m, p_j$
7.1	$p_j, ..., p_k, ..., p_i, ..., p_m$	$?, ?$	$p_k, p_i : p_m, p_j$
7.2	$p_j, ..., p_k, ..., p_i, ..., p_m$	$?, ?$	$p_k, p_i : p_m, p_j$

Для случаев 2-4 кажется важным вопрос о выборе точек (какая p_i, какая p_j), однако на самом деле изменение индексации приводит к тому, что случай начинает восприниматься как другой, но имеющий такую же реакцию. Это значит, что выбор точек во всех случаях может быть случаен, а из таблицы можно выкинуть две пары:

номер	порядок в массиве	правильность	порядок в отрезке
1.1	$..., p_j, p_i, ...$	$+$	p_j, p_i
1.2	$..., p_i, p_j, ...$	$+$	p_i, p_j
2.1	$p_j, ..., p_i, ..., p_m, ..., p_k$	$-, -$	$p_i, p_k : p_j, p_m$
2.2	$p_i, ..., p_j, ..., p_m, ..., p_k$	$?, -$	$p_i, p_k : p_j, p_m$
3.1	$p_j, ..., p_i, ..., p_m, ..., p_k$	$-, -$	$p_k, p_i : p_j, p_m$
3.2	$p_i, ..., p_j, ..., p_m, ..., p_k$	$?, ?$	$p_k, p_i : p_j, p_m$