COMP 3011
DESIGN AND ANALYSIS OF ALGORITHMS
FALL 2024

Randomized Algorithms

LI Bo
Department of Computing
The Hong Kong Polytechnic University

COMMENTS

- Review some basic math concepts or the previous contents when encountered.
- I have never learned calculus before, so I don't understand some of the expressions or notation. Maybe explain the notation first, instead of skipping steps.

  Sure.

- You could add some problems with answers left for us to exercise.

  Sure. Will prepare exercise problems for final.

RANDOMIZED ALGORITHMS

RANDOM VARIABLES

RANDOM VARIABLES

- Let X be a ***discrete random variable***.
- In particular, for every real number a, there is some value $\Pr[X = a]$ that says what is the total probability of all events where X takes value a. These values satisfy:

$$\Pr[X = a] \geq 0 \text{ and } \sum_a \Pr[X = a] = 1$$

- Saying that X is discrete means $\Pr[X = a] > 0$ for only finitely (or countably) many values a.

Definition (***Expected Value, Expectation, Mean***)

$$E[X] = \sum_a a \cdot \Pr[X = a]$$

Proposition (Probabilistic Method)

- There is some outcome such that X takes value $\geq E[X]$.
- There is some outcome such that X takes value $\leq E[X]$.

RANDOM VARIABLES

We can construct new random variables from old ones.

- For example, if X and Y are random variables then so to is $X + Y$.
- It is the random variable that takes the value of X plus the value of Y on each outcome.

Proposition (***Linearity of Expectation***)

For two random variables X and Y, over the same probability space, we have

$$E[X + Y] = E[X] + E[Y].$$

Furthermore, for a random variable X and a constant α we have

$$E[\alpha \cdot X] = \alpha \cdot E[X].$$

RANDOM VARIABLES

Definition

joint probability of $X = a$ and $Y = b$

X and Y are ***independent*** if for all values,

$$\Pr[X = a] \cdot \Pr[Y = b] = \Pr[X = a \text{ and } Y = b]$$

Proposition

If X and Y are independent, then $E[X \cdot Y] = E[X] \cdot E[Y]$.

More generally, if $X_1, \ldots, X_n$ are independent (meaning any two of them are independent) then

$$E[\Pi_i X_i] = \Pi_i E[X_i].$$

MARKOV'S INEQUALITY

Definition

Say that a random variable is nonnegative if $\Pr[X = a] > 0$ only for $a \geq 0$.

Theorem (Markov's Inequality)

- If X is a nonnegative random variable, then for any $\alpha > 0$ we have $\Pr\left[X \geq \alpha \cdot E[X]\right] \leq \frac{1}{\alpha}$.
- Equivalently, $\Pr[X \geq \alpha] \leq \frac{E[X]}{\alpha}$.

Proof.

by setting α to be $\alpha \cdot E[X]$.

The first statement follows immediately from the second statement.

$$E[X] = \sum_{a \geq \alpha} a \cdot \Pr[X = a] + \sum_{a < \alpha} a \cdot \Pr[X = a] \geq \sum_{a \geq \alpha} \alpha \cdot \Pr[X = a] = \alpha \cdot \Pr[X \geq \alpha]$$

UNION BOUND

- Sometimes we want to avoid a collection of bad events that may not be independent.
- The probability that some bad event happens is upper bounded by the sum of the individual probabilities of the bad events.

Theorem (Union Bound)

Consider any collection $X_1, X_2, \ldots, X_n$ of $\{0,1\}$ random variables. Then

$$\Pr[X_i = 1 \text{ for some } 1 \leq i \leq n] \leq \sum_{i=1,\ldots,n} \Pr[X_i = 1].$$

LAW OF TOTAL PROBABILITY

***Theorem* (*Law of Total Probability*)**

If $\{B_1, \dots, B_n\}$ is a finite (or countably infinite) partition of a sample space (in other words, a set of pairwise disjoint events whose union is the entire sample space), then for any event A of the same probability space

joint probability:
the probability of A happens and B_i happens

$$\Pr[A] = \sum_{i=1}^{n} \Pr[A \cap B_i]$$

or, alternatively,

conditional probability:
the probability of A happens, given B_i happens

$$\Pr[A] = \sum_{i=1}^{n} \Pr[\, A \mid B_i \,] \cdot \Pr[B_i] \,.$$

WAITING FOR A FIRST SUCCESS

- We have a coin: come up head with probability $p > 0$, and tail $1 - p$.
- Different flips have independent outcomes.
- We flip the coin until we first get a head. What's the expected number of flips we perform?
- Let X be the random variable equal to the number of flips performed.
- For $j > 0$, we have $\Pr[X = j] = (1 - p)^{j-1}p$:

  The first $j - 1$ flips must come up tails, and the j-th must come up head. Thus

$$E[X] = \sum_{j=1}^{\infty} j \cdot \Pr[X = j] = \sum_{j=1}^{\infty} j \cdot (1 - p)^{j-1}p = \frac{1}{p}$$

Theorem (Waiting for a First Success)

If we repeatedly perform independent trials of an experiment, each of which succeeds with probability $p > 0$, then the expected number of trials we need to perform until the first success is $\frac{1}{p}$.

THE MAX-SAT PROBLEM

THE MAX-SAT PROBLEM

- In the maximum satisfiability problem (MAX-SAT), we are given clauses $C_1, \dots, C_m$, each a disjunction of literals over variables $x_1, \dots, x_n$ (e.g. $C_j = (x_1 \vee \overline{x_2} \vee x_3)$).
- Each of the variables x_i may be set to either true or false. The objective of the problem is to find a truth assignment that satisfies the maximum possible number of clauses.

The optimization version of SAT

Algorithm 1 (Flipping Coins)

Independently for each i, set $x_i = \begin{cases} true, \ with\ probability \frac{1}{2} \\ false, \ with\ probability \frac{1}{2} \end{cases}$

THE MAX-SAT PROBLEM

Lemma

For each clause C with, say, k literals,

$$\Pr[C \text{ is satisfied }] \geq 1 - \frac{1}{2^k}.$$

Proof: Instead of computing $\Pr[C \text{ is satisfied}]$, we compute $\Pr[C \text{ is not satisfied}]$.

$$\Pr[C \text{ is not satisfied}] = \Pi_{x_i \in C} \Pr[x_i \text{ is false}] \cdot \Pi_{\overline{x}_i \in C} \Pr[x_i \text{ is true}] = \left(\frac{1}{2}\right)^k$$

follows since x_i's are sampled independently.

Corollary

For MAX 3-SAT problem and any clause C,

$$\Pr[C \text{ is satisfied }] \geq \frac{7}{8}.$$

THE MAX-3SAT PROBLEM

α–approximation randomized algorithm:
$E[ALG] \geq \alpha \cdot OPT$

Theorem

For MAX 3-SAT problem, the expected number of satisfied clauses is at least $\frac{7}{8}m$, where m is the number of clauses.

$E[ALG1] \geq \frac{7}{8}m \geq \frac{7}{8}OPT$: Algorithm 1 is $\frac{7}{8}$–approximation

Proof:

$$E[\#\text{ satisfied clauses}] = \sum_C \Pr[C \text{ is satisfied}] * 1 \geq \frac{7}{8}m\,.$$

➢ For any random variable, there must be some point at which it assumes some value at least as large as its expectation.

Theorem

For every instance of 3-SAT, there is a truth assignment that satisfies at least a $\frac{7}{8}$ fraction of all clauses.

PROBABILISTIC METHOD

Theorem

For every instance of 3-SAT, there is a truth assignment that satisfies at least a $\frac{7}{8}$ fraction of all clauses.

- We have arrived at a nonobvious fact about 3-SAT:

  The existence of an assignment satisfying many clauses, whose statement has nothing to do with randomization; but we have done so by a randomized construction.

- This is a fairly widespread principle in the area of combinatorics:

  One can show the existence of some structure by showing that a random construction produces it with positive probability.

- Constructions of this sort are said to be applications of the ***probabilistic method***.

THE MAX-3SAT PROBLEM

- Suppose we are not satisfied with a "one-shot" algorithm that produces a single assignment with a large number of satisfied clauses in expectation.
- Rather, we would like a randomized algorithm whose ***expected running time is polynomial*** and that is ***guaranteed to output a truth assignment satisfying at least a $\frac{7}{8}$ fraction of all clauses***.
- A simple way to do this is to generate random truth assignments until one of them satisfies at least $\frac{7}{8}m$ clauses.
- How long will it take until we find one by random trials?

Waiting for a First Success?

THE MAX-3SAT PROBLEM

- If we can show that the probability a random assignment satisfies at least $\frac{7}{8}m$ clauses is at least p, then the expected number of trials performed by the algorithm is $\frac{1}{p}$.
- What is this quantity p?
- For $j = 1, \dots, m$, let p_j denote the probability that a random assignment satisfies exactly j clauses.
- So the expected number of clauses satisfied, by the definition of expectation, is equal to

$$\sum_{j=1}^{m} j \cdot p_j$$

By the previous analysis, this is equal to $\frac{7}{8}m$.

- We are interested in the quantity $p = \sum_{j \geq \frac{7}{8}m} p_j$.

THE MAX-3SAT PROBLEM

$$\frac{7}{8}m = \sum_{j=1}^{m} j \cdot p_j \quad = \sum_{j<\frac{7}{8}m} j \cdot p_j + \sum_{j\geq\frac{7}{8}m} j \cdot p_j$$

- If we can show that the probability a random assignment satisfies at least $\frac{7}{8}m$ clauses is at least p, then the expected number of trials performed by the algorithm is $\frac{1}{p}$.

- Let k' denote <u>the largest natural number that is strictly smaller</u> than $\frac{7}{8}m$.
- The right-hand side of the above equation only increases if we replace the terms in the first sum by $k'p_j$ and the terms in the second sum by mp_j. We have

$$\frac{7}{8}m = \sum_{j<\frac{7}{8}m} jp_j + \sum_{j\geq\frac{7}{8}m} jp_j \leq \sum_{j<\frac{7}{8}m} k'p_j + \sum_{j\geq\frac{7}{8}m} mp_j = k'(1-p) + mp \leq k' + mp$$

THE MAX-3SAT PROBLEM

$$\frac{7}{8}m \leq k' + mp \quad \Rightarrow \quad mp \geq \frac{7}{8}m - k'$$

- Since k' is a natural number strictly smaller than $\frac{7}{8}$ times another natural number,

$$\frac{7}{8}m - k' \geq \frac{1}{8}$$

- Thus,

$$p \geq \frac{\frac{7}{8}m - k'}{m} \geq \frac{1}{8m}.$$

- By the waiting-time bound, we see that the expected number of trials needed to find the satisfying assignment we want is at most $8m$.

THE MAX-3SAT PROBLEM

Summary: Improvement of Algorithm 1 (Flipping Coins)

- Repeat Algorithm 1 until we research a correct solution: ***Las Vegas Algorithms***
  - A Las Vegas algorithm is a randomized algorithm whose output is always correct.
  - The expected running time of the algorithm is polynomial.

- Repeat Algorithm 1 polynomial times: ***Monte Carlo Algorithms***
  - A Monte Carlo algorithm is a randomized algorithm whose output may be incorrect with a certain (typically small) probability.
  - The running time of the algorithm is always polynomial.

DE-RANDOMIZATION

DE-RANDOMIZATION

Law of Total Probability:

$$\Pr[A] = \sum_{i=1}^{n} \Pr[\, A \mid B_i \,] \cdot \Pr[B_i]\,.$$

$E[\text{\# satisfied clauses}]$

$$= E[\,\text{\#satisfied clauses} \mid x_1 = true\,] \cdot \Pr[x_1 = true] +$$

$$E[\,\text{\#satisfied clauses} \mid x_1 = true\,] \cdot \Pr[x_1 = false]$$

$$= \frac{1}{2}(E[\,\text{\#satisfied clauses} \mid x_1 = true\,] + E[\,\text{\#satisfied clauses} \mid x_1 = false\,])$$

$$\max \begin{cases} E[\,\text{\#satisfied clauses} \mid x_1 = true\,] \\ E[\,\text{\#satisfied clauses} \mid x_1 = false\,] \end{cases} \geq E[\text{\# satisfied clauses}] \geq \frac{7}{8}m$$

If $E[\,\text{\#satisfied clauses} \mid x_1 = true\,] \geq E[\,\text{\#satisfied clauses} \mid x_1 = false\,]$, set $x_1 = true$;

If $E[\,\text{\#satisfied clauses} \mid x_1 = true\,] < E[\,\text{\#satisfied clauses} \mid x_1 = false\,]$, set $x_1 = false$

DE-RANDOMIZATION

Suppose we set $x_1 = b_1$

$$E[\,\#\text{satisfied clauses} \mid x_1 = b_1\,] \longrightarrow \geq E[\#\text{ satisfied clauses}] \geq \frac{7}{8}m$$

$$= E[\,\#\text{satisfied clauses} \mid x_1 = b_1, x_2 = true\,] \cdot \Pr[x_2 = true] +$$

$$E[\,\#\text{satisfied clauses} \mid x_1 = b_1, x_2 = true\,] \cdot \Pr[x_2 = false]$$

$$= \frac{1}{2}(E[\,\#\text{satisfied clauses} \mid x_1 = b_1 x_1 = true\,] +$$

$$E[\,\#\text{satisfied clauses} \mid x_1 = b_1 x_1 = false\,])$$

$$\max \begin{cases} E[\,\#\text{satisfied clauses} \mid x_1 = b_1, x_2 = true\,] \\ E[\,\#\text{satisfied clauses} \mid x_1 = b_1 x_1 = false\,] \end{cases} \geq E[\,\#\text{satisfied clauses} \mid x_1 = b_1\,]$$

DE-RANDOMIZATION

Suppose we set $x_1 = b_1$

$$E[\,\#\text{satisfied clauses} \mid x_1 = b_1\,] \longrightarrow \geq E[\#\text{ satisfied clauses}] \geq \frac{7}{8}m$$

$$= E[\,\#\text{satisfied clauses} \mid x_1 = b_1, x_2 = true\,] \cdot \Pr[x_2 = true] +$$

$$E[\,\#\text{satisfied clauses} \mid x_1 = b_1, x_2 = true\,] \cdot \Pr[x_2 = false]$$

$$= \frac{1}{2}(E[\,\#\text{satisfied clauses} \mid x_1 = b_1 x_1 = true\,] +$$

$$E[\,\#\text{satisfied clauses} \mid x_1 = b_1 x_1 = false\,])$$

If $E[\,\#\text{satisfied clauses} \mid x_1 = b_1, x_2 = true\,] \geq E[\,\#\text{satisfied clauses} \mid x_1 = b_1, x_2 = false\,]$,

set $x_2 = true$;

If $E[\,\#\text{satisfied clauses} \mid x_1 = b_1, x_2 = true\,] < E[\,\#\text{satisfied clauses} \mid x_1 = b_1, x_2 = false\,]$,

set $x_2 = false$.

DE-RANDOMIZATION

$$\max \begin{cases} E[\,\#\text{satisfied clauses} \mid x_1 = b_1, \dots, x_i = b_i, x_{i+1} = true\,] \\ E[\,\#\text{satisfied clauses} \mid x_1 = b_1, \dots, x_i = b_i, x_{i+1} = false\,] \end{cases} \geq E[\#\text{satisfied clauses}] \geq \frac{7}{8}m$$

Suppose we have set $x_1 = b_1, \dots, x_i = b_i$.

Case 1: If $E[\,\#\text{satisfied clauses} \mid x_1 = b_1, \dots, x_i = b_i, x_{i+1} = true\,] \geq$

$E[\,\#\text{satisfied clauses} \mid x_1 = b_1, \dots, x_i = b_i, x_{i+1} = false\,]$,

set $x_{i+1} = true$;

Case 2: If $E[\,\#\text{satisfied clauses} \mid x_1 = b_1, \dots, x_i = b_i, x_{i+1} = true\,] <$

$E[\,\#\text{satisfied clauses} \mid x_1 = b_1, \dots, x_i = b_i, x_{i+1} = false\,]$,

set $x_{i+1} = false$.

COMPUTATIONAL GEOMETRY

COMPUTATIONAL GEOMETRY

Our questions:

- How to represent a point in a 2D plane?
- How to represent a line (segment) in a 2D plane?

Further, we also have the following questions:

- How to determine whether two line segments intersect?
- How to determine whether there are two line segments in a given set of segments intersect?

GEOMETRIC OBJECTS

Some simple geometric objects in the 2D plane can be represented as:

- ***Point***: A point can be simply represented by a pair of real numbers $p = (x, y)$ that correspond to its coordinates. It can also be interpreted by a vector from $(0, 0)$ to (x, y).
- ***Line***: A line may be represented by two points (x_1, y_1) and (x_2, y_2) on it, or, more efficiently, by a pair of real numbers, namely, the ***slope*** k and the y-***intercept*** b, i.e.,

$$y = kx + b, \text{ where } k = \frac{y_2 - y_1}{x_2 - x_1}.$$

- ***Line Segment***: A line segment can be represented by its two endpoints $\overline{p_0 p_1}$ and $\overrightarrow{p_0 p_1}$ if it is directed.

GEOMETRIC OBJECTS

Some simple geometric objects in the 2D plane can be represented as:

- ***Linear Combination***: The Linear Combination of $p_1, p_2, \cdots, p_n \in \mathbb{R}^d$ is defined as:

$$\left\{\sum_i \alpha_i \cdot p_i : \alpha_i \in \mathbb{R}\right\}$$

- ***Affine Combination***: The Affine Combination of $p_1, p_2, \cdots, p_n \in \mathbb{R}^d$ is defined as:

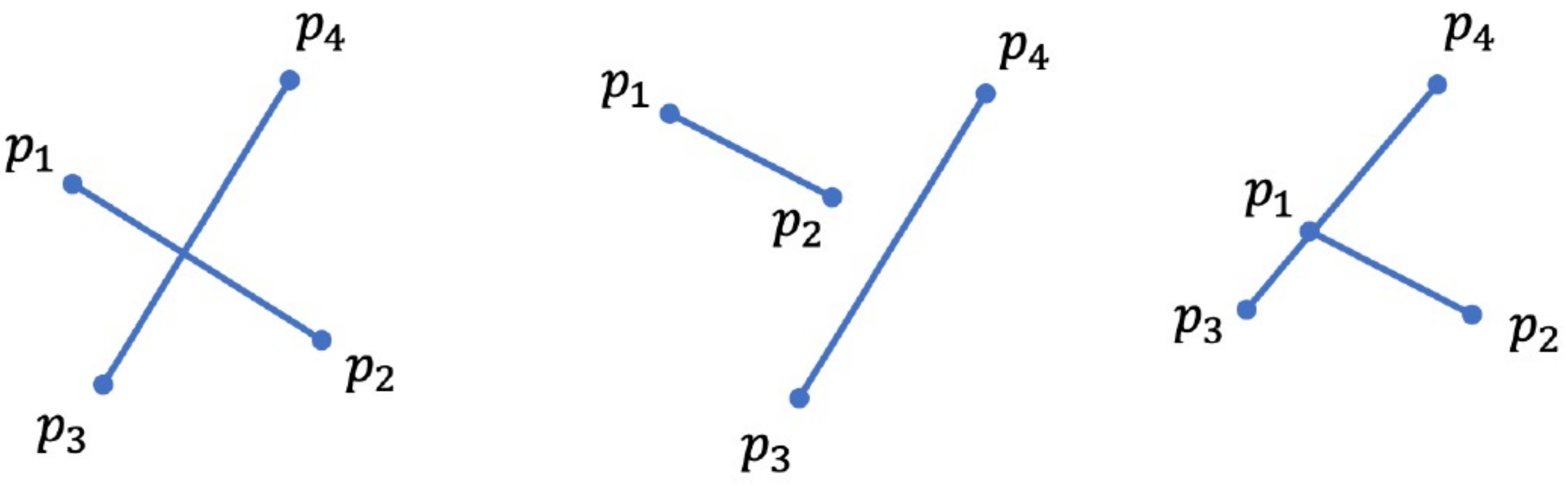
$$\left\{\sum_i \alpha_i \cdot p_i : \sum_i \alpha_i = 1\right\}$$

- ***Convex Combination***: The Convex Combination of $p_1, p_2, \cdots, p_n \in \mathbb{R}^d$ is defined as:

$$\left\{\sum_i \alpha_i \cdot p_i : \alpha_i \geq 0 \text{ and } \sum_i \alpha_i = 1\right\}$$

How to determine whether two line segments intersect?

GEOMETRIC ALGORITHMS



- A segment p_1p_2 ***straddles*** a line if point p_1 lies on one side of the line and point p_2 lies on the other side.
- A boundary case arises if p_1 or p_2 lies directly on the line.
- Two line segments intersect if and only if either of the following conditions holds:
  - Each segment straddles the line containing the other.
  - An endpoint of one segment lies on the other segment.

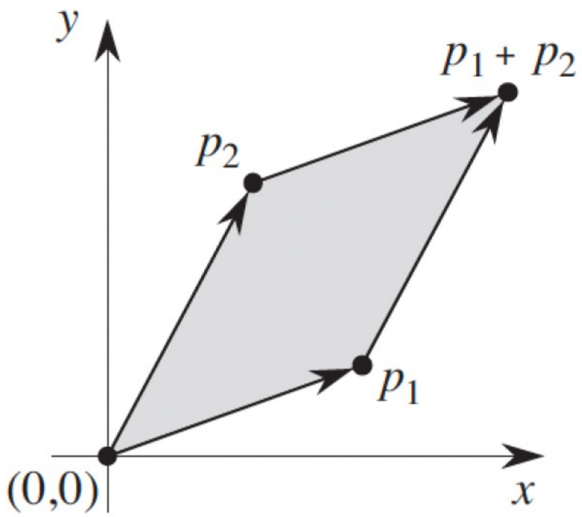
CROSS PRODUCTS

Consider vectors $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$.

- We can interpret the ***cross product*** $p_1 \times p_2$ as the signed area of the parallelogram formed by the points $(0,0), p_1, p_2, p_1 + p_2 = (x_1 + x_2, y_1 + y_2)$.
- A more useful definition of the cross-product is

$$p_1 \times p_2 = x_1 y_2 - x_2 y_1 = -p_2 \times p_1$$

- If $p_1 \times p_2$ is positive, p_1 is ***clockwise*** from p_2 with respect to the origin;
- If this cross product is negative, then p_1 is ***counterclockwise*** from p_2.
- A boundary condition arises if the cross product is 0; in this case, the vectors are ***colinear***, pointing in either the same or opposite directions.



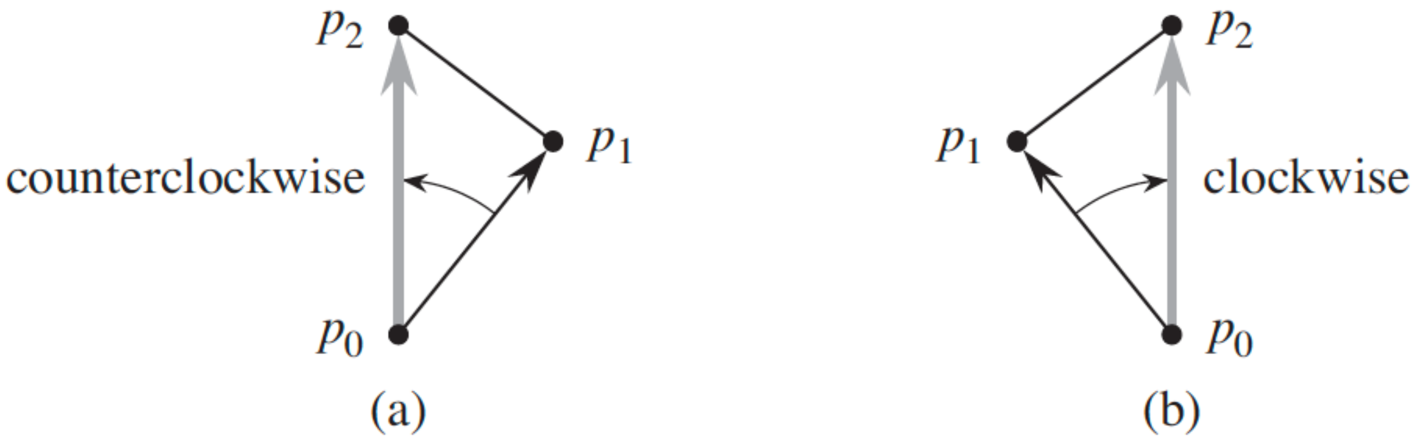
CROSS PRODUCTS

- What if the common endpoint is not $(0,0)$?

  How to determine whether a directed segment $\overrightarrow{p_0p_1}$ is closer to $\overrightarrow{p_0p_2}$ in a clockwise direction or in a counterclockwise direction with respect to their common endpoint p_0?

- We can simply consider alternative vectors $p_1' = p_1 - p_0$ and $p_2' = p_2 - p_0$, and check the cross product of p_1' and p_2'.

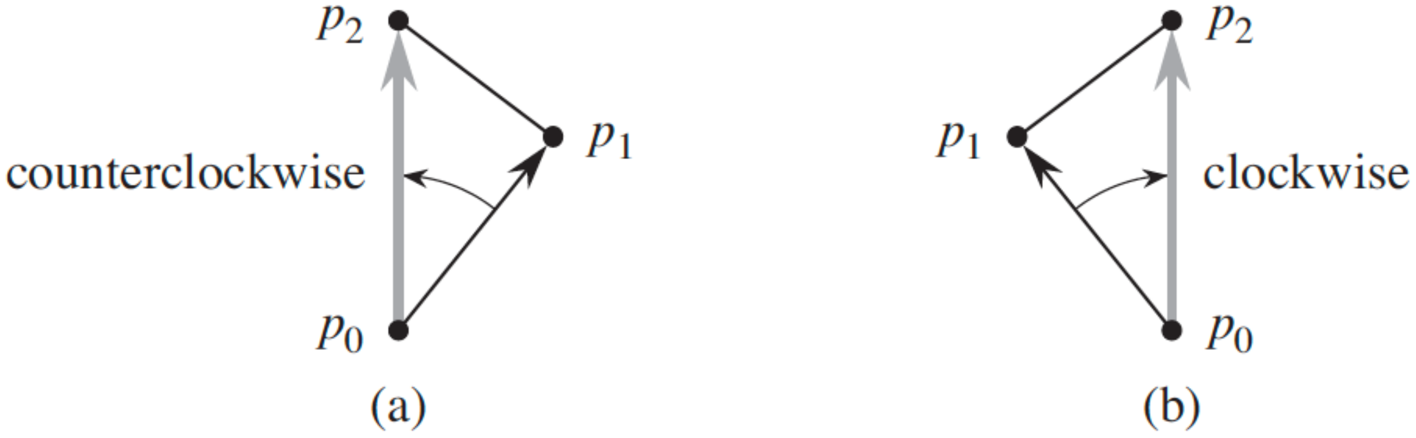
$$(p_1 - p_0)\times(p_2 - p_0) = (x_1 - x_0)(y_2 - y_0) - (x_2 - x_0)(y_1 - y_0)$$



Whether Consecutive Segments Turn Left or Right?

Whether two consecutive line segments $\overline{p_0p_1}$ and $\overline{p_0p_2}$ turn left or right at point p_1?

- We can compute the cross product $(p_2 - p_0)\times(p_1 - p_0)$.
- If the sign of this cross-product is negative, $\overline{p_0p_2}$ is counterclockwise with respect to $\overline{p_0p_1}$ and thus we make a left turn at p_1.
- A positive cross-product indicates a clockwise orientation and a right turn.
- A cross product of 0 means that points p_0, p_1 and p_2 are colinear.

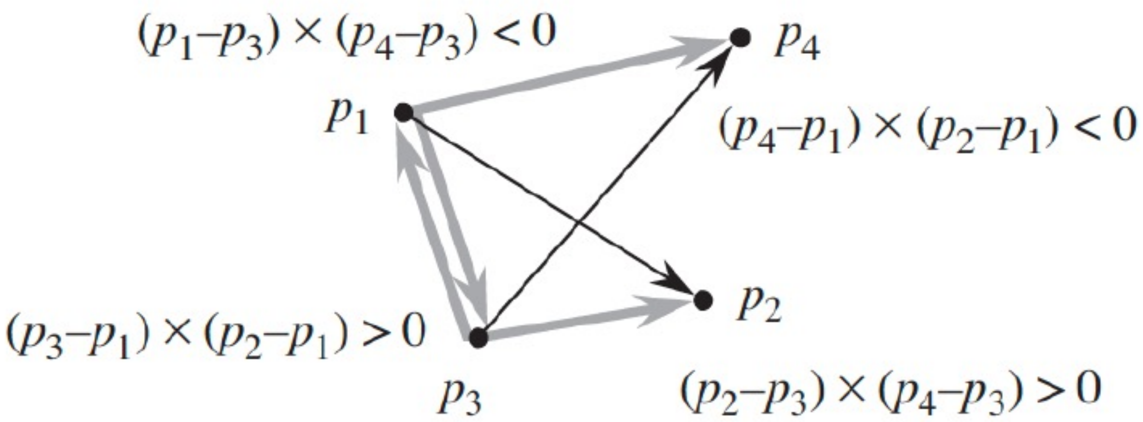


Determine Whether Two Line Segments Intersect

- A segment $\overline{p_1 p_2}$ straddles a line if point p_1 lies on one side of the line and point p_2 lies on the other side.
- If the segments straddle each other, they intersect.

We can use cross-product to determine the orientations.

- Segment $\overline{p_1 p_2}$ straddles the line containing segment $\overline{p_3 p_4}$ if directed segments $\overrightarrow{p_3 p_1}$ and $\overrightarrow{p_3 p_2}$ have opposite orientations relative to $\overrightarrow{p_3 p_4}$.
- Similarly, $\overline{p_3 p_4}$ straddles the line containing $\overline{p_1 p_2}$ if directed segments $\overrightarrow{p_1 p_3}$ and $\overrightarrow{p_1 p_4}$ have opposite orientations relative to $\overrightarrow{p_1 p_2}$.

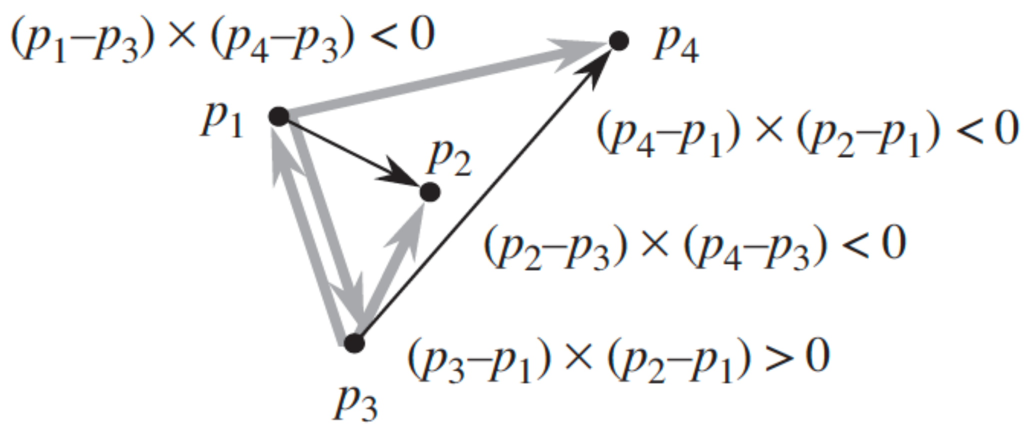


Determine Whether Two Line Segments Intersect

- A segment $\overline{p_1p_2}$ straddles a line if point p_1 lies on one side of the line and point p_2 lies on the other side.
- If the segments straddle each other, they intersect.

We can use cross-product to determine the orientations.

- Segment $\overline{p_1p_2}$ straddles the line containing segment $\overline{p_3p_4}$ if directed segments $\overrightarrow{p_3p_1}$ and $\overrightarrow{p_3p_2}$ have opposite orientations relative to $\overrightarrow{p_3p_4}$.
- Similarly, $\overline{p_3p_4}$ straddles the line containing $\overline{p_1p_2}$ if directed segments $\overrightarrow{p_1p_3}$ and $\overrightarrow{p_1p_4}$ have opposite orientations relative to $\overrightarrow{p_1p_2}$.

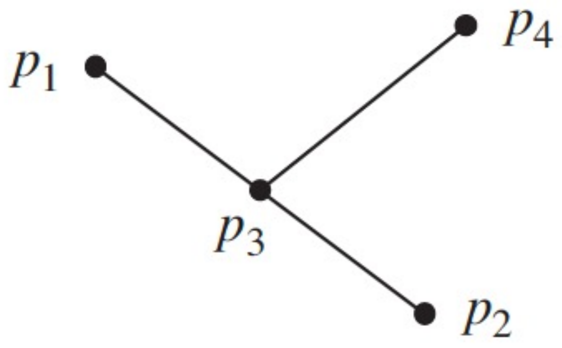


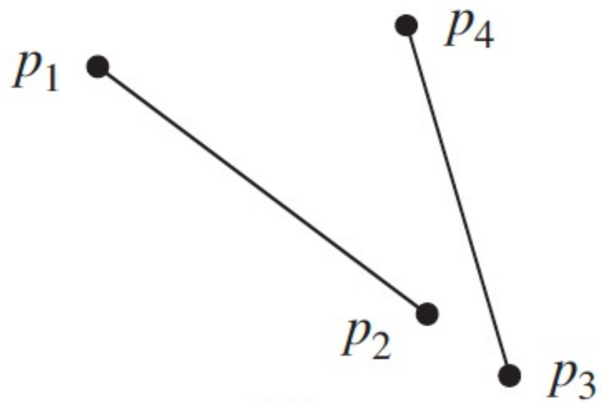
Determine Whether Two Line Segments Intersect

But we have ***Boundary Case***:

- There is some p_k that is colinear with the other segment (the cross-product is 0).
- It is directly on the other segment if and only if it is between the endpoints of the other segment.
- For example, if p_3 is colinear with p_1p_2, we only need to check

$$\min\{x_1, x_2\} \le x_3 \le \max\{x_1, x_2\} \text{ and } \min\{y_1, y_2\} \le y_3 \le \max\{y_1, y_2\}$$





Conclusion: Determine whether two line segments intersect can be done in $O(1)$ time.