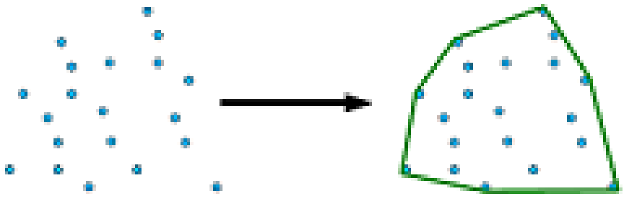
GeeksforGeeks

A computer science portal for geeks

GeeksQuiz

Convex Hull | Set 1 (Jarvis's Algorithm or Wrapping)

Given a set of points in the plane. the convex hull of the set is the smallest convex polygon that contains all the points of it.



We strongly recommend to see the following post first.
How to check if two given line segments intersect?

The idea of Jarvis's Algorithm is simple, we start from the leftmost point (or point with minimum x coordinate value) and we keep wrapping points in counterclockwise direction. The big question is, given a point p as current point, how to find the next point in output? The idea is to use orientation() here. Next point is selected as the point that beats all other points at counterclockwise orientation, i.e., next point is q if for any other point r, we have "orientation(p, r, q) = counterclockwise". Following is the detailed algorithm.

1) Initialize p as leftmost point.
2) Do following while we don't come back to the first (or leftmost) point.
…..**a)** The next point q is the point such that the triplet (p, q, r) is counterclockwise for any other point r.
…..**b)** next[p] = q (Store q as next of p in the output convex hull).
…..**c)** p = q (Set p as q for next iteration).

```
// A C++ program to find convex hull of a set of points
// Refer http://www.geeksforgeeks.org/check-if-two-given-line-segments-inters
// for explanation of orientation()
#include <iostream>
using namespace std;

// Define Infinite (Using INT_MAX caused overflow problems)
#define INF 10000

struct Point
{
    int x;
    int y;
};

// To find orientation of ordered triplet (p, q, r).
// The function returns following values
// 0 --> p, q and r are colinear
// 1 --> Clockwise
// 2 --> Counterclockwise
int orientation(Point p, Point q, Point r)
{
    int val = (q.y - p.y) * (r.x - q.x) -
              (q.x - p.x) * (r.y - q.y);

    if (val == 0) return 0;  // colinear
    return (val > 0)? 1: 2; // clock or counterclock wise
}

// Prints convex hull of a set of n points.
void convexHull(Point points[], int n)
{
    // There must be at least 3 points
    if (n < 3) return;

    // Initialize Result
    int next[n];
    for (int i = 0; i < n; i++)
        next[i] = -1;
    // Find the leftmost point
    int l = 0;
    for (int i = 1; i < n; i++)
        if (points[i].x < points[l].x)
            l = i;

    // Start from leftmost point, keep moving counterclockwise
    // until reach the start point again
    int p = l, q;
    do
    {
        // Search for a point 'q' such that orientation(p, i, q) is
        // counterclockwise for all points 'i'
        q = (p+1)%n;
        for (int i = 0; i < n; i++)
          if (orientation(points[p], points[i], points[q]) == 2)
             q = i;

        next[p] = q;  // Add q to result as a next point of p
        p = q; // Set p as q for next iteration
    } while (p != l);

    // Print Result
    for (int i = 0; i < n; i++)
    {
        if (next[i] != -1)
           cout << "(" << points[i].x << ", " << points[i].y << ")\n";
    }
}

// Driver program to test above functions
int main()
{
    Point points[] = {{0, 3}, {2, 2}, {1, 1}, {2, 1},
                      {3, 0}, {0, 0}, {3, 3}};
    int n = sizeof(points)/sizeof(points[0]);
    convexHull(points, n);
    return 0;
}
```

Output: The output is points of the convex hull.

```
(0, 3)
(3, 0)
(0, 0)
(3, 3)
```

Time Complexity: For every point on the hull we examine all the other points to determine the next point. Time complexity is [Tex]\Theta(m * n)[/Tex] where n is number of input points and m is number of output or hull points (m <= n). In worst case, time complexity is O(n ^{2}). The worst case occurs when all the points are on the hull (m = n)

We will soon be discussing other algorithms for finding convex hulls.

Sources:
http://www.cs.uiuc.edu/~jeffe/teaching/373/notes/x05-convexhull.pdf
http://www.dcs.gla.ac.uk/~pat/52233/slides/Hull1x1.pdf

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

Related Topics:

- Linearity of Expectation
- Iterative Tower of Hanoi
- Count possible ways to construct buildings
- Build Lowest Number by Removing n digits from a given number
- Set Cover Problem | Set 1 (Greedy Approximate Algorithm)
- Find number of days between two given dates
- How to print maximum number of A's using given four keys
- Write an iterative O(Log y) function for pow(x, y)

Tags: geometric algorithms, MathematicalAlgo

Writing code in comment? Please use **ideone.com** and share the link here.

14 Comments GeeksforGeeks 1 Login

Recommend Share Sort by Newest

Join the discussion…

Deepesh Maheshwari · 5 months ago

q = (p+1)%n;

Why this is used to get next point, didn't get the logic behind it

Reply · Share ›

Ram ➦ Deepesh Maheshwari · 4 months ago

if the left most x point, i.e. 'p' in the example is the last element of the array, to get the next point, i.e. the 0th element, we need to do (p+1)%n which will return 0.

Reply · Share ›

ggm8 · 5 months ago

This doesn't output them in counter-clockwise order... It just finds them in a random order.

3 · Reply · Share ›

poname · 7 months ago

not works well for
9 points

x	y
0	0
0	1
1	0
0	2
2	0
0	3
3	0
1	2
2	1

output must be 9 but gives 6

2 ⌃ | ⌄ • Reply • Share ›

mayank · a year ago

in the line "... point is q if for any other point r, we have “orientation(p, r, q) = counterclockwise. Following is the detailed algorithm......." it should be..."orientation(p, q, r) = counterclockwise"...

@admin please check this

1 ⌃ | ⌄ • Reply • Share ›

Deepesh Maheshwari ↪ mayank · 5 months ago

It is p,r,q - I also confused with it.
just run the program in debug mode, you will get the explanation

⌃ | ⌄ • Reply • Share ›

michael_skynet · a year ago

Thanks, helped me a lot :)

⌃ | ⌄ • Reply • Share ›

Tyler Johnsson · a year ago

Fantastic article, thank you very much!
I just wanted to share that I spent a good few hours tearing my hair over why it only worked "sometimes". Then I realized that it had to be p = l (lima) and not p = 1 (one) as I originally thought, haha. Thank you again!

1 ⌃ | ⌄ • Reply • Share ›

venkat · a year ago

for points

⌃ | ⌄ • Reply • Share ›

Manish Kumar • 2 years ago

There is one more error. The for loop in both cases should start from 0 and not from 1.

^ | ⌄ • Reply • Share ›

> **GeeksforGeeks** ➦ Manish Kumar • 2 years ago
>
> @Manish Kumar: Thanks for your inputs. We have updated the second loop. The first loop looks fine though, it's a typical way to find min value in an array.
>
> 1 ^ | ⌄ • Reply • Share ›

vsethuooo • 2 years ago

Inside the convexHull(..) function,
inside the do while loop,
why do you use, [script]for(i=1;i<n;i++)[script]="" instead="" of="" [script]for(i="0;i&lt;n;i++)
[/script]" ??="" pls="" explain="">

^ | ⌄ • Reply • Share ›

sumit1294 • 2 years ago

Time complexity of the given algo. must be O(m x n) not O(m x m) as mentioned above.

1 ^ | ⌄ • Reply • Share ›

> **GeeksforGeeks** ➦ sumit1294 • 2 years ago
>
> Thanks for pointing this out. This was a typo. We have corrected it. Keep it up!
>
> 2 ^ | ⌄ • Reply • Share ›

✉ Subscribe   Add Disqus to your site   Privacy   DISQUS

Google™ Custom Search

-
-
-
-   - Interview Experiences
    - Advanced Data Structures

  - Dynamic Programming
  - Greedy Algorithms
  - Backtracking
  - Pattern Searching
  - Divide & Conquer
  - Mathematical Algorithms
  - Recursion
  - Geometric Algorithms

Popular Posts

- All permutations of a given string
- Memory Layout of C Programs
- Understanding “extern” keyword in C
- Median of two sorted arrays
- Tree traversal without recursion and without stack!
- Structure Member Alignment, Padding and Data Packing
- Intersection point of two Linked Lists
- Lowest Common Ancestor in a BST.
- Check if a binary tree is BST or not
- Sorted Linked List to Balanced BST

Follow @GeeksforGeeks

Recent Comments

- radek

  hey venki..is there a way to implement the same...

  The Ubiquitous Binary Search | Set 1 · 1 minute ago

- radek

  hey..is there a way to implement the same...

  The Ubiquitous Binary Search | Set 1 · 2 minutes ago

- Nikhil kumar

  public class...

  Print missing elements that lie in range 0 – 99 · 35 minutes ago

- Ashish Aggarwal

  Try Data Structures and Algorithms Made Easy -...

  Algorithms · 57 minutes ago

- Vlad

  Thanks. Very interesting lectures.

  Expected Number of Trials until Success · 2 hours ago

- cfh

  My implementation which prints the index of the...

  Longest Even Length Substring such that Sum of First and Second Half is same · 2 hours ago

-