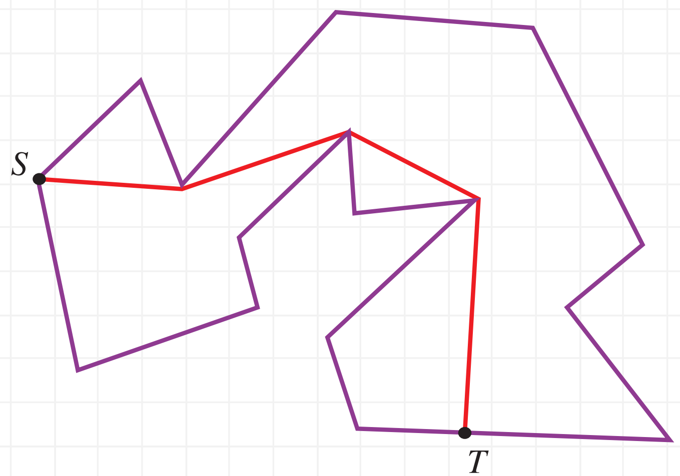
Basic geometric shortest path algorithms — shortest paths in 2D

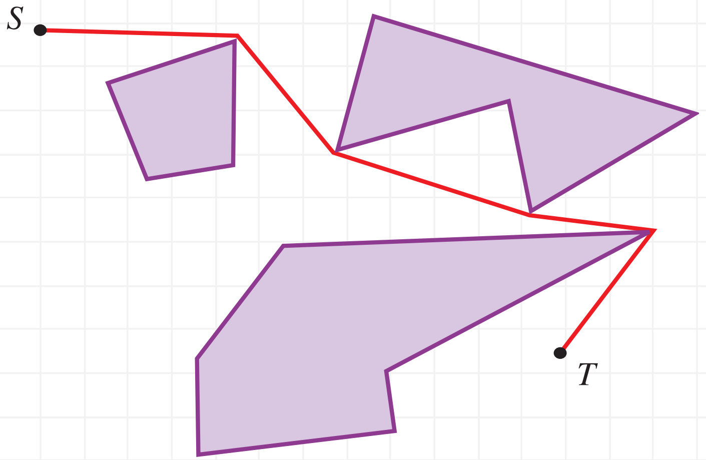
polygon



$n = \#$ vertices

$O(n)$

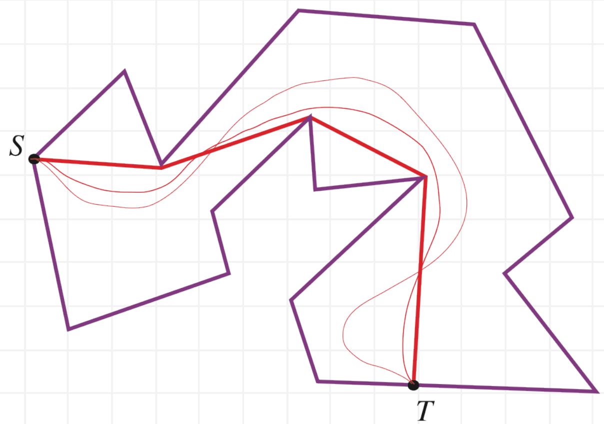
polygonal domain
polygon with holes



$\Theta(n \log n)$

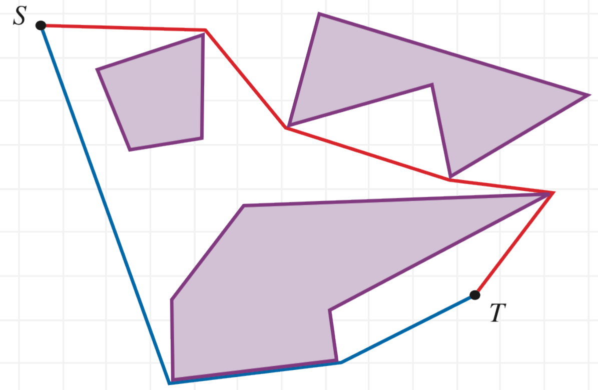
Basic geometric shortest path algorithms — shortest paths in 2D

polygon



unique

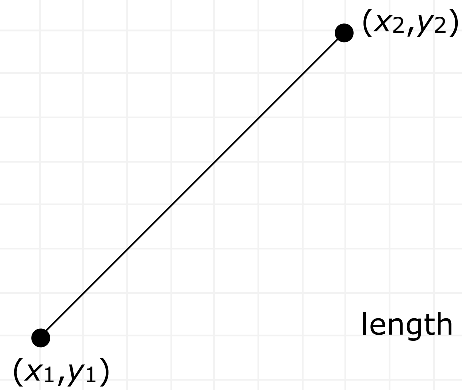
polygonal domain



locally shortest paths

not unique

Basic geometric shortest path algorithms — shortest paths in 2D



length = $\sqrt{(x_2 - x_1)^2 + (y_2 - x_1)^2}$

No one knows a polynomial time algorithm for this question:

Given a path of line segments, is the length $\leq t$?

problem: sum of square roots

no known poly. bound on # bits

to decide $\sum \sqrt{a_i} \geq k$

Assume *real RAM* model of computation.

square root computations at unit cost.

Basic geometric shortest path algorithms — shortest paths in 2D polygon

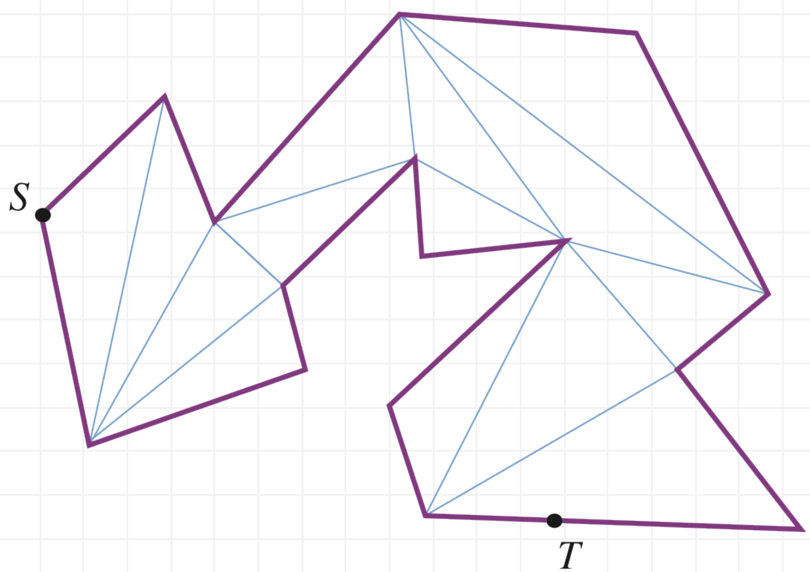
Given a polygon, two points *S*, *T*, find the shortest path from *S* to *T*

Funnel algorithm

$O(n)$

[Euclidean shortest paths in the presence of rectilinear barriers](http://scholar.google.ca/scholar?q=lee+preparata&btnG=&hl=en&as_sdt=0%2C5)
DT **Lee**, FP **Preparata** - Networks, 1984 - Wiley Online Library
Cited by 345 Related articles All 3 versions Cite Save

From: http://scholar.google.ca/scholar?q=lee+preparata&btnG=&hl=en&as_sdt=0%2C5

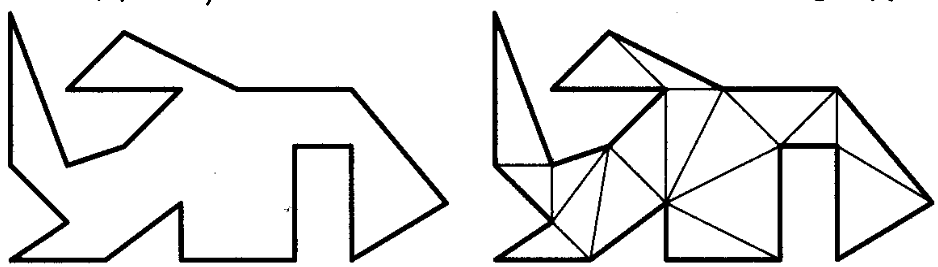


First find a triangulation in linear time
— not implementable

Aside: Triangulations

Fact: every simple polygon of n vertices has a triangulation with n-3 *chords.*

Pf. by induction. Just show $\exists$ one chord.



Daniel Vlasic

Good O(n log n) time algorithms.

O(n) time algorithm, but not implementable:

[**Triangulating** a simple polygon in **linear time**]
B **Chazelle** - Discrete & Computational Geometry, 1991 - Springer
Cited by 805Related articlesAll 18 versionsCiteSave
From: http://scholar.google.ca/scholar?q=chazelle+linear+time+triangulation&btnG=&hl=en&as_sdt=0%2C5

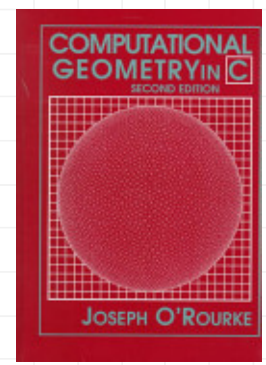
O(n) time randomized algorithm: to present

[**A randomized algorithm** for **triangulating a simple polygon** in **linear time**]
NM Amato, MT Goodrich, EA Ramos - Discrete & Computational Geometry, 2001 - Springer
Cited by 21Related articlesAll 4 versionsCiteSave
From: http://scholar.google.ca/scholar?hl=en&q=A+Randomized+Algorithm+for+Triangulating+a+Simple+Polygon+in+Linear+Time&btnG=&as_sdt=1%2C5&as_sdtp=

Fact: every polygonal region of n vertices and h holes has a triangulation with n+h-3 *chords.*

computational geometry books

http://www.cs.uu.nl/geobook/

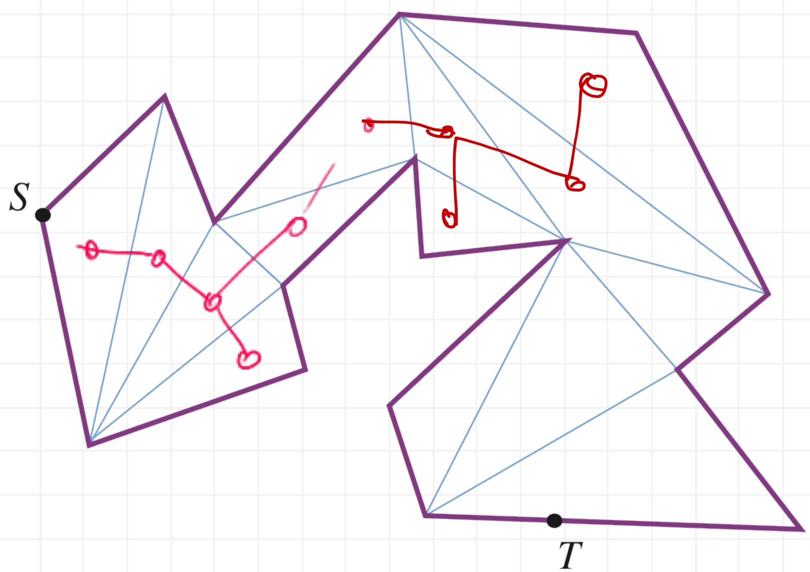


http://books.google.ca/books/about/Computational_G

Basic geometric shortest path algorithms — shortest paths in 2D polygon

Given a polygon, two points S, T, find the shortest path from S to T

Funnel algorithm

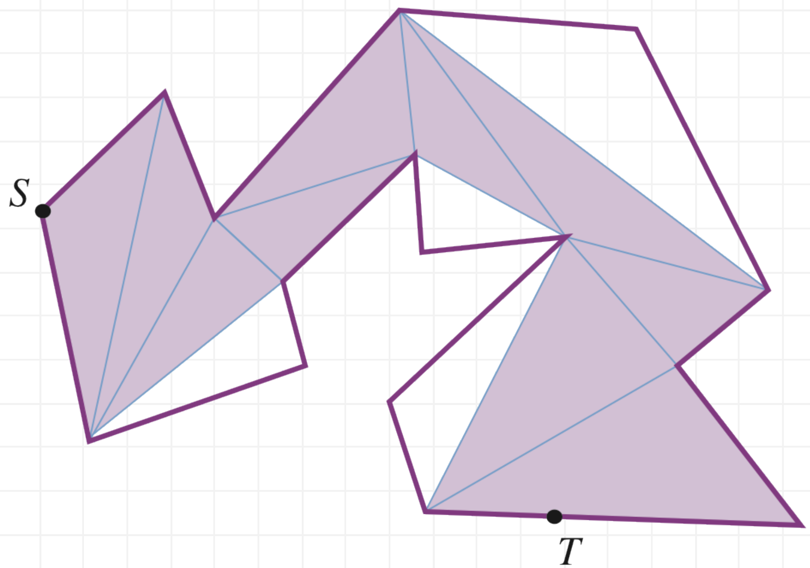


the dual of the triangulation is a tree

Basic geometric shortest path algorithms — shortest paths in 2D polygon

Given a polygon, two points S, T, find the shortest path from S to T

Funnel algorithm

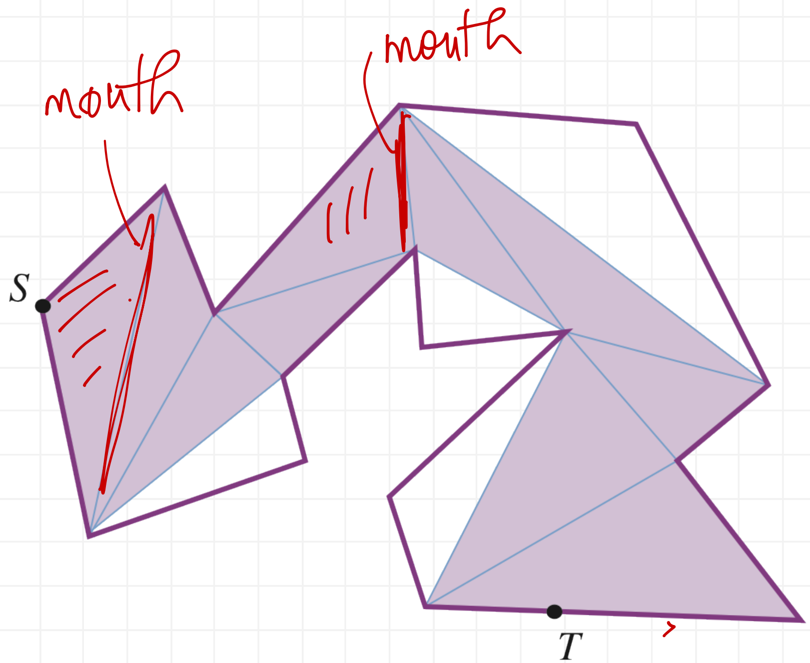


find the path of triangles from S to T

Basic geometric shortest path algorithms — shortest paths in 2D polygon

Given a polygon, two points S, T, find the shortest path from S to T

Funnel algorithm



go along the path of triangles, maintaining all shortest paths from S leaving the *mouth* of the current triangle

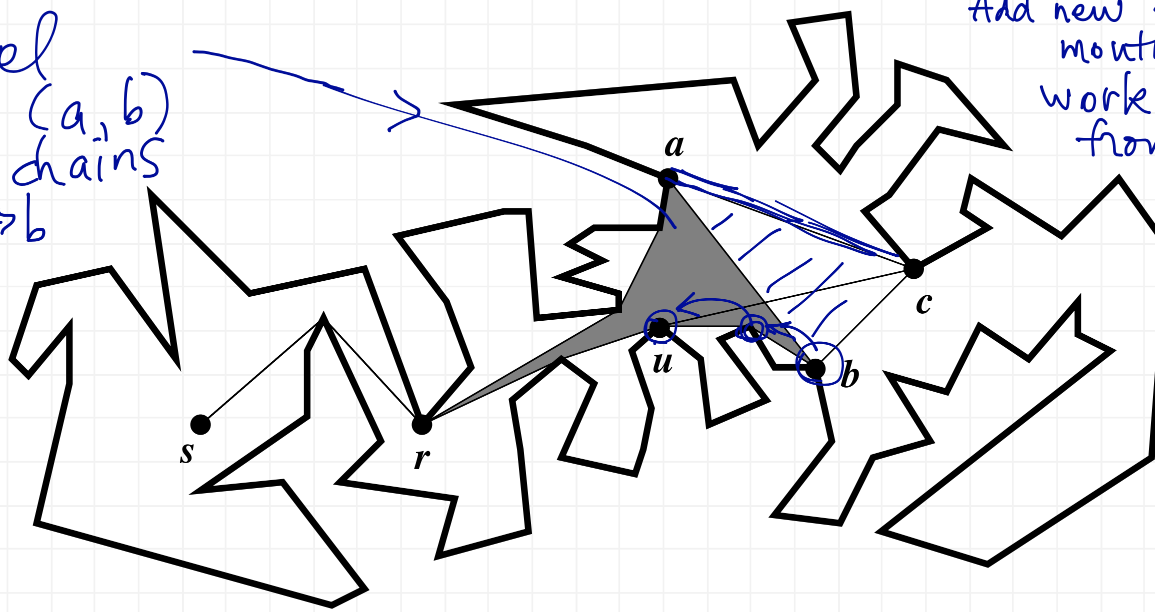
Basic geometric shortest path algorithms — shortest paths in 2D polygon

Given a polygon, two points S, T, find the shortest path from S to T

Funnel algorithm, general step

funnel
mouth (a, b)
2 reflex chains
$r \to a$, $r \to b$

Add new Δ abc
mouth a, c
work backwards from b to recover reflex chain.



Joseph Mitchell

Running time

Each Δ may cause $O(n)$ update.

But each vertex is discarded only once so $O(n)$ total

Basic geometric shortest path algorithms — shortest paths in 2D polygon

query version: Given a polygon, point *S* preprocess to handle query: given *T*, find the shortest path from *S* to *T.*

measure P, S, Q — query time

space

preprocessing time

$P \in O(n)$

$S \in O(n)$

$Q \in O(\log n)$ – for length

$O(\log n + k)$ – for actual path

$k = \#$ vertices on path

Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons
L Guibas, J Hershberger, D Leven, M Sharir, RE Tarjan - Algorithmica, 1987 - Springer
From: http://scholar.google.ca/scholar?hl=en&q=Linear-Time+Algorithms+for+Visibility+and+Shortest+Path+Problems+Inside+Triangulated+Simple+Polygons&btnG=&as_sdt=1%2C5&as_sdtp=

[HTML] Computing minimum length paths of a given homotopy class
J Hershberger, J Snoeyink - Computational geometry, 1994 - Elsevier
Cited by 140Related articlesAll 2 versionsWeb of Science: 62CiteSave

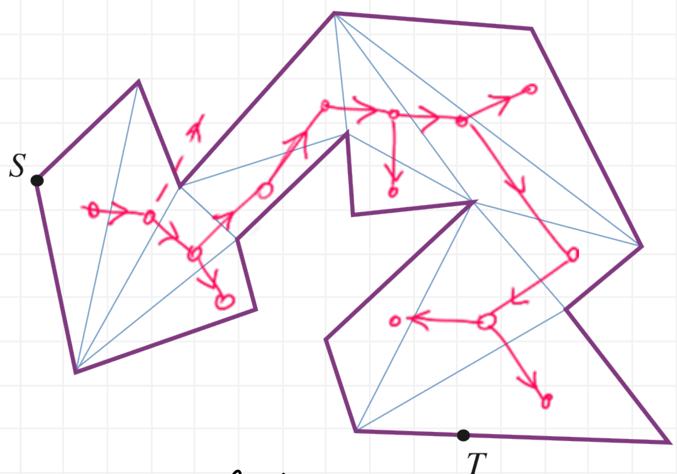
From: http://scholar.google.ca/scholar?q=Computing+minimum+length+paths+of+a+given+homotopy+class&btnG=&hl=en&as_sdt=0%2C5

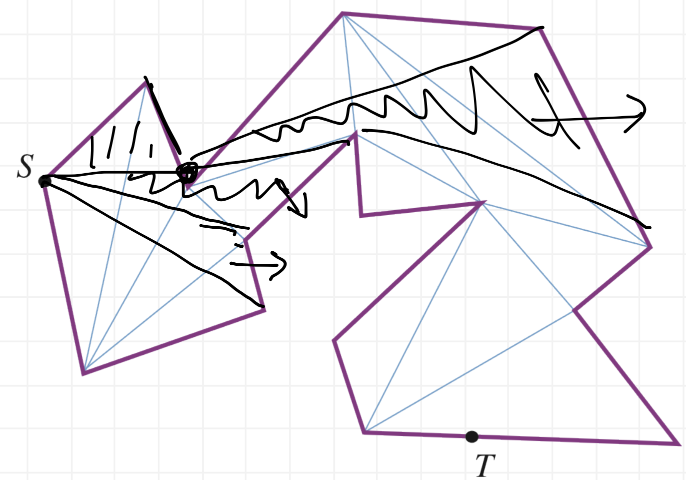
simpler data structure

Basic geometric shortest path algorithms — shortest paths in 2D polygon

Given a polygon, point *S* preprocess to handle query: given *T*, find the shortest path from *S* to *T*

Idea of algorithm

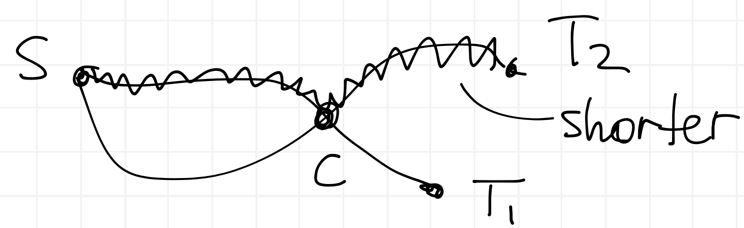




traverse dual tree
root (at S) to leaves

find funnels to all polygon edges

implicit: shortest paths from S
do not cross



Issues

how to split funnels efficiently

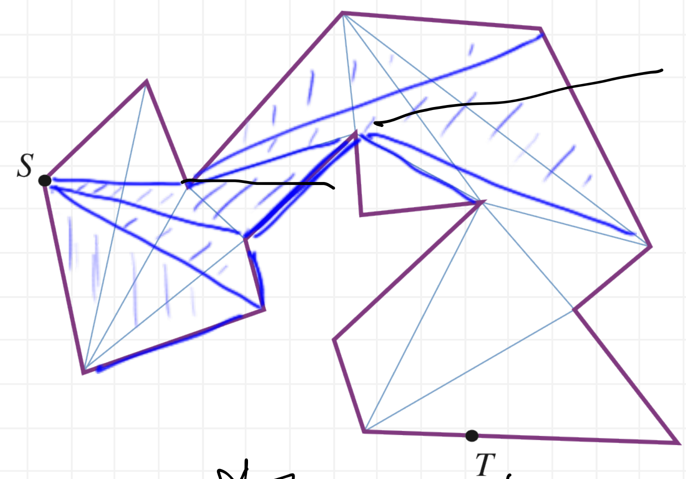
what if S is not a vertex?

what if query point T is not a vertex?

Basic geometric shortest path algorithms — shortest paths in 2D polygon

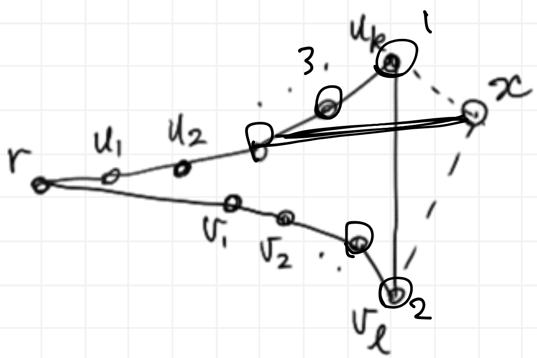
Given a polygon, point *S* preprocess to handle query: given *T*, find the shortest path from *S* to *T*

Idea of algorithm



Issues

how to split funnels efficiently



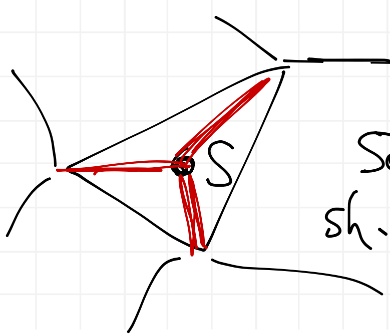
store funnel of dequeue $u_k \cdots u_1 \, r \, v_1 \cdots v_\ell$

discard from both ends. Linear search $O(n \log n)$

Binary search $O(n)$. No details

Then use planar point location to find region containing query T.

what if S is not a vertex?

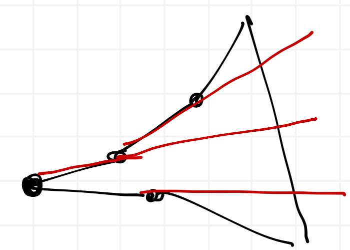


Solve 3 sh. path problems

what if query point T is not a vertex?

Subdivide funnels further to get Shortest Path Map

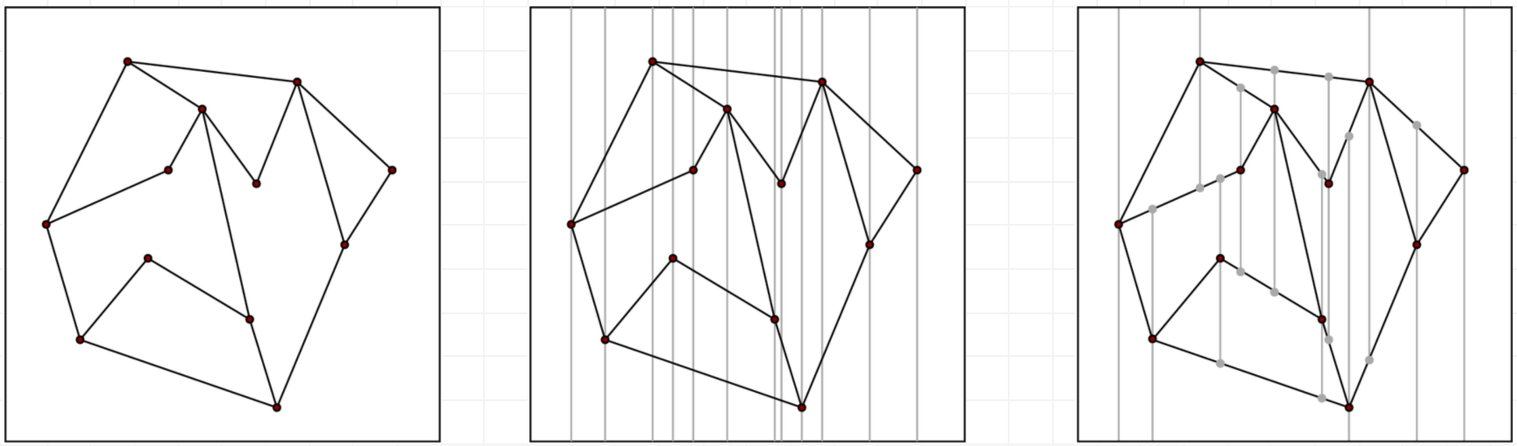
— in each region, all pts. have same vertex seq. for shortest path



Aside: Planar Point Location

Fact: Given a straight-line planar decomposition of size *n*, can preprocess in time O(*n*), to data structure of size O(*n*) to handle query of the following type in time O(log *n*): given a point *p*, which region is it in?

http://en.wikipedia.org/wiki/Point_location



slab method (easy)

$P = S = O(n^2)$

$Q = O(\log n)$

trapezoidal decomposition

– practical

most randomized

Seidel

Basic geometric shortest path algorithms — shortest paths in 2D polygon

2-Point Query Version: Given a polygon, preprocess to handle query: given *S*, *T*, find the shortest path from *S* to *T*.

P = S = O(n)
Q = O(log n + k)

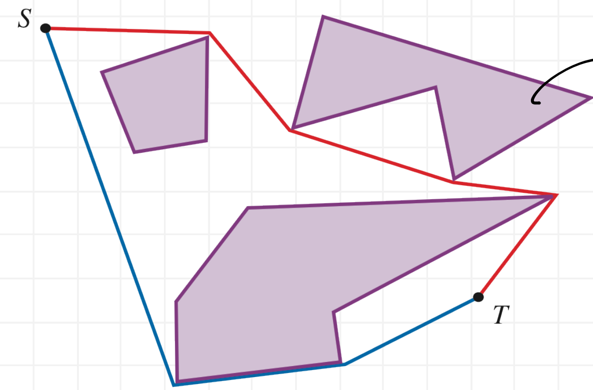
[HTML] Optimal shortest path queries in a simple polygon
LJ Guibas, J Hershberger - Journal of Computer and System Sciences, 1989 - Elsevier
From: http://scholar.google.ca/scholar?cluster=8433806075988061775&hl=en&as_sdt=0,5

A **new data structure** for **shortest path queries** in a **simple polygon**
J Hershberger - Information Processing Letters, 1991 - Elsevier
From: http://scholar.google.ca/scholar?q=A+new+data+structure+for+shortest+path+queries+in+a+simple+polygon&btnG=&hl=en&as_sdt=2005&sciodt=0%2C5&cites=8433806075988061775&scipsc=

could present.

Basic geometric shortest path algorithms — shortest paths in 2D polygonal domain

Given a polygonal domain, two points S, T, find the shortest path from S to T



hole

holes = h

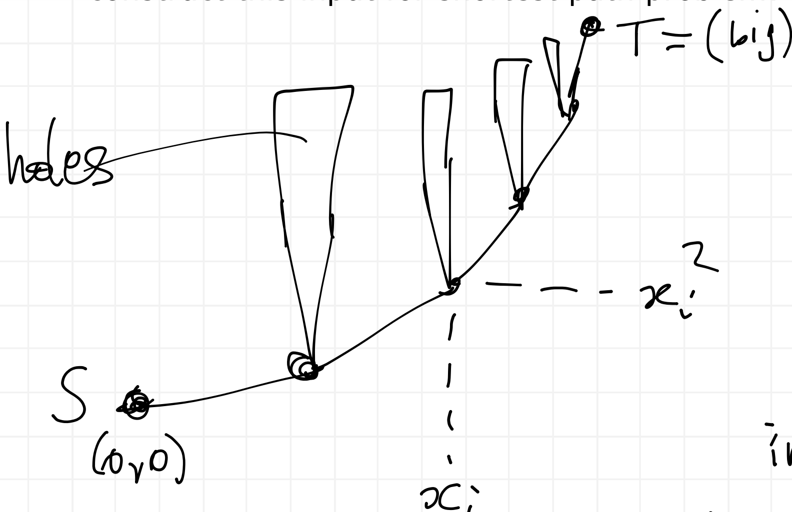
vertices = n

↑ total

Why n log n is a lower bound if we can only compare numbers

input n numbers to sort: $x_1 \ldots x_n$

construct this input for shortest path problem:



x_i — positive integers

create hole for x_i at (x_i, x_i^2) (going up)

The shortest path $S \to T$ gives sorted order of x_i's

So $\Omega(n \log n)$ for sh. paths in comparison model (+ random access)

Actually $\Omega(h \log h)$

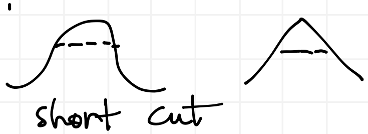
Basic geometric shortest path algorithms — shortest paths in 2D polygonal domain

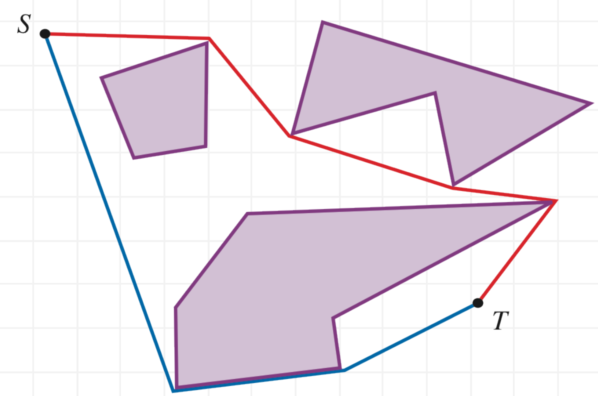
Given a polygonal domain, two points S, T, find the shortest path from S to T

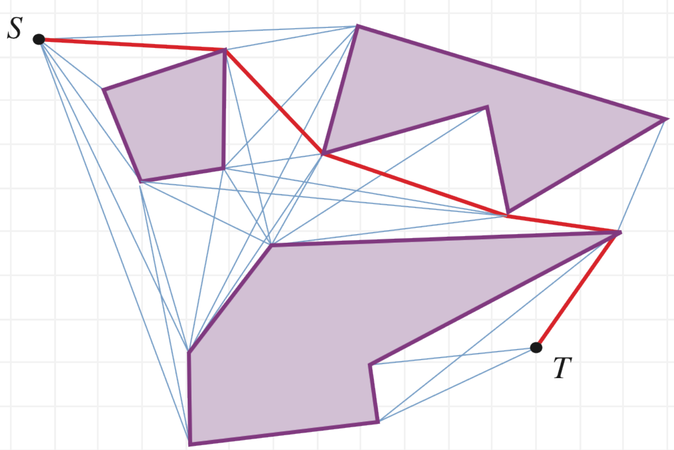
Reducing to a graph problem

Claim. The shortest path uses segments that join pairs of vertices of the polygonal domain.

Pf.







Algorithm:

construct the *visibility graph*

apply Dijkstra's algorithm

add edge (u, v) if line segment (u,v) does not enter any hole

$w(u,v)$ = length of segment

Basic geometric shortest path algorithms — shortest paths in 2D polygonal domain

Given a polygonal domain, two points *S*, *T*, find the shortest path from *S* to *T*

Reducing to a graph problem

Algorithm:

construct the *visibility graph*

apply Dijkstra's algorithm

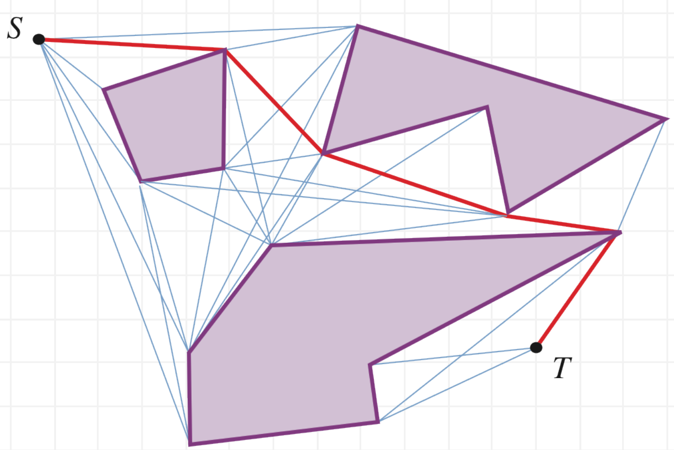
Run time $O(m + n \log n)$ $m = \#$ edges — can be n^2

[An output-sensitive algorithm for computing visibility graphs](http://scholar.google.ca/scholar?q=An+output-sensitive+algorithm+for+computing+visibility+graphs&btnG=&hl=en&as_sdt=0%2C5)

SK Ghosh, DM Mount - SIAM Journal on **Computing**, 1991 - SIAM

Cited by 191 Related articles All 4 versions Web of Science: 73 Cite Save

From: http://scholar.google.ca/scholar?q=An+output-sensitive+algorithm+for+computing+visibility+graphs&btnG=&hl=en&as_sdt=0%2C5



vis. graphs are dense

OPEN: test (in poly. time) if a graph is a visibility graph (of a simple polygon).