NATIONAL ICT AUSTRALIA

COMP5045 Computational Geometry

NATIONAL ICT AUSTRALIA

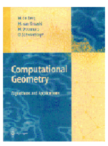
COMP5045 – Computational Geometry

Course page:
http://www.gudmundsson.biz/comp5045/

Lecturers:
Joachim Gudmundsson & Thomas Wolle
NICTA, School of IT Building, Level 5 West

NATIONAL ICT AUSTRALIA

Course book:
M. de Berg, M. Overmars, M. van Kreveld and O. Schwarzkopf
Computational Geometry: Algorithms and Application, Springer-Verlag



Outline:
13 lectures, Thursday 14:00-16:00

Assignments:
3 assignments, emailed as pdf-file (preferably using latex)
Handing in late is not accepted!

Grade:
Each assignment is 25% of your grade
Final exam is 25% of your grade (you must not fail the exam)

Collaboration:
General ideas - Yes!
Formulation and writing - No!

The imagination driving Australia's ICT future.

NATIONAL ICT AUSTRALIA

Brief History: Algorithms and CG

Algorithms:
300 BC: Euclid designed an algorithm for GCD.
780 AD: al-Khowarizmi – the word "Algorithm" is derived from his name.
…
Babbage, Cantor, Hilbert, Church, Gödel
…
1936: Turing developed a machine that provides a formal and rigorous definition of an algorithm

Computational Geometry:
1644: Descartes wrote about Voronoi diagrams
1759: Euler & Vandermonde discussed Euclidean TSP
1978: Shamos wrote his thesis which defines the start of modern CG
1985: Preparata & Shamos wrote the first CG textbook

4

The imagination driving Australia's ICT future.

NATIONAL ICT AUSTRALIA

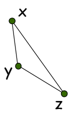
What is computational geometry?

The study of algorithms to solve problems stated in terms of geometry.

The problems we study are defined in a metric space!

For every two points x and y in the metric space, there is a function $d(x,y) \geq 0$ which gives the distance between them as a nonnegative real number. A metric space must also satisfy

1. $d(x,y) = 0$ iff $x = y$,
2. $d(x,y) = d(y,x)$,
3. The triangle inequality must hold $d(x,y) + d(y,z) \geq d(x,z)$.



We will mainly consider the Euclidean metric (L_2-metric).

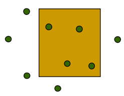
5

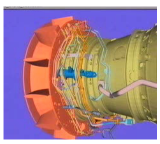
The imagination driving Australia's ICT future.

NATIONAL ICT AUSTRALIA

Why computational geometry?

Certain problems are inherently geometric:
- Point location
- Range searching
- Motion planning
- …



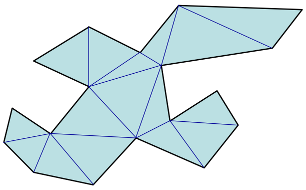


Other problems are just much easier to solve if we use the underlying metric:
- Travelling Salesman Problem
- Nearest neighbour
- …

6

Computational Geometry

Polygon Triangulation and The Art Gallery Problem

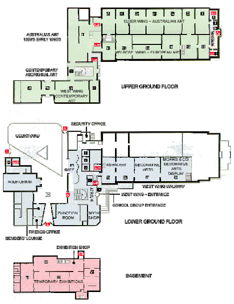


The Art Gallery problem

Question:

How many guards are needed to guard an art gallery?

Victor Klee posed this problem to Václav Chvátal in 1973.

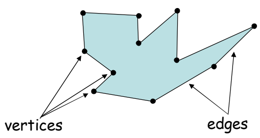


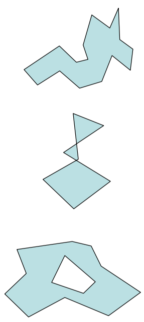
Art Gallery - definitions

Input: An Art Gallery =
A simple polygon with n line segments

Polygon: A region of the plane bounded by a set of straight line segments forming a closed curve.

Simple: no self-intersection, no holes



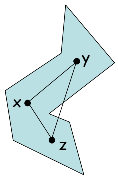


Art Gallery

Guard: (camera, motion sensors, fire detectors, …)

- 2π range visibility
- infinite distance
- cannot see through walls
- cannot move

point a can see point b
iff $(a,b) \subseteq P$

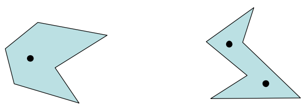


The imagination driving Australia's ICT future.

Art Gallery

Question: How many guards are needed to guard an art gallery?

n=6



G(n) = the smallest number of guards that suffice to guard any polygon with n vertices.

Is $G(n) \leq n$? If we place one guard on each vertex?

The imagination driving Australia's ICT future.

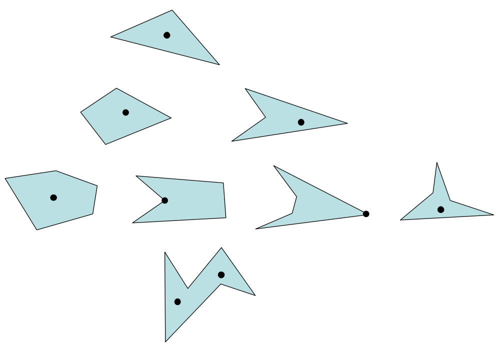
A lower bound

n=3

n=4

n=5

n=6



The imagination driving Australia's ICT future.

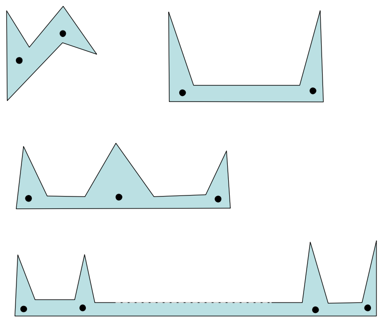
NATIONAL ICT AUSTRALIA

A lower bound

n=6

n=9

n=3k



13

The imagination driving Australia's ICT future.

NATIONAL ICT AUSTRALIA

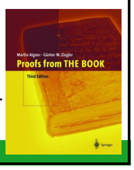
A conjecture

We have shown: $G(n) \geq \lfloor n/3 \rfloor$

Conjecture: $G(n) = \lfloor n/3 \rfloor$
Proved by Chvátal in 1975.

"Steve Fisk learned of Klee's question from Chvátal's article, but found the proof unappealing. He continued thinking about the problem and came up with a solution while dozing off on a bus travelling somewhere in Afghanistan."

An elegant proof was given by Fisk in 1978.
Included in "Proofs from the book".



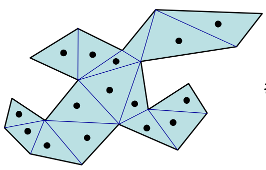
The imagination driving Australia's ICT future.

NATIONAL ICT AUSTRALIA

A first upper bound

Prove that $G(n) \leq n-2$.

- Decompose P into pieces that are easy to guard. For example triangles!
- How can we use a triangulation of P to place a set of guards?



#guards = #triangles

15

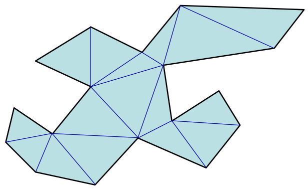
The imagination driving Australia's ICT future.

NATIONAL ICT AUSTRALIA

Triangulation

A triangulation can be obtained by adding a maximal number of non-intersecting diagonals within P.

A diagonal of P is a line segment between two vertices of P that are clearly visible to each other.



BUT!

1. Does a triangulation always exist?
2. What is the number of triangles?

The imagination driving Australia's ICT future.

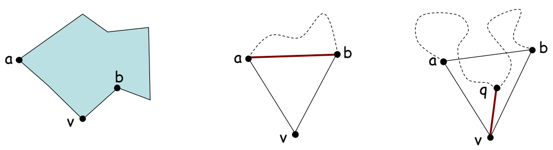
NATIONAL ICT AUSTRALIA

Number of triangles

Does a triangulation always exist?

Is there always a diagonal?

Lemma: Every simple polygon with >3 vertices has a diagonal.



The imagination driving Australia's ICT future.

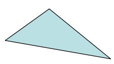
NATIONAL ICT AUSTRALIA

Number of triangles

Theorem: Every simple polygon admits a triangulation.

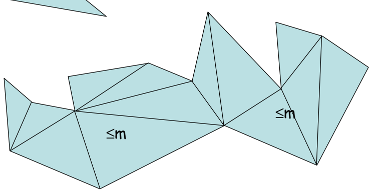
Proof by induction (over the number of vertices):

Base case:
n=3



Induction hyp.:
n≤m

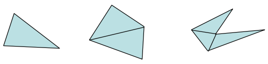
Induction step:
n=m+1



Number of triangles

Theorem: Every triangulation of P of n vertices consists of x triangles.

What's x?

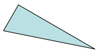


Conjecture: x = n-2

Number of triangles

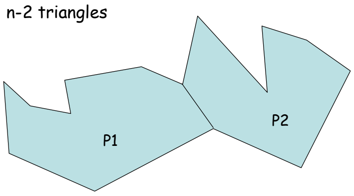
Proof by induction:

Base case (n = 3):



Ind. hyp. (n ≤ m): n-2 triangles

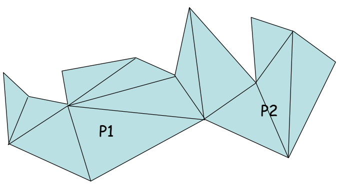
Ind. step (n = m+1):



Number of triangles

|P1| = m1
|P2| = m2

m1+m2 = n + 2

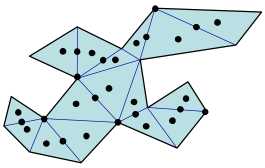


According to ind. hyp.
#triangles in P1 is m1-2
#triangles in P2 is m2-2

(m1-2) + (m2-2) = m1 + m2 - 4 = n – 2 QED

Theorem: Every triangulation of P consists of n-2 triangles.

Back to the Art Gallery



$G(n) \geq \lfloor n/3 \rfloor$

$G(n) \leq$ #guards = #triangles = n-2

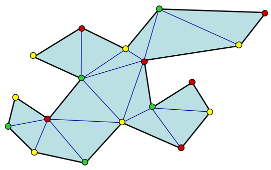
How do we place the guards?

The imagination driving Australia's ICT future.

3-colouring

Idea: Assign a colour to each vertex such that no two adjacent vertices have the same colour.

#yellow = 7
#green = 5
#red = 6



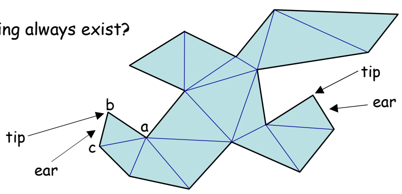
Place guards on the green vertices.

$\Rightarrow$ #guards $\leq \lfloor n/3 \rfloor$

The imagination driving Australia's ICT future.

3-colouring

Does a 3-colouring always exist?



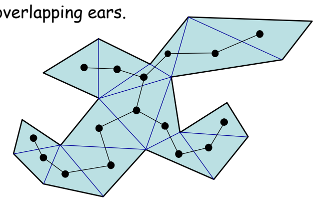
Definition: Three consecutive vertices a, b and c of a triangulation form an ear of P if ac is a diagonal of P, where b is the ear tip.

NATIONAL ICT AUSTRALIA

3-colouring

Theorem: Every polygon with n>3 vertices has at least two non-overlapping ears.

Consider the dual D(T) of the triangulation T.



D(T) is a binary tree.

Every tree with at least two nodes has at least two nodes of degree 1 $\Rightarrow$ T has at least two ears.

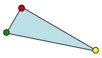
NATIONAL ICT AUSTRALIA

3-colouring

Theorem: The triangulation of a simple polygon can always be 3-coloured.

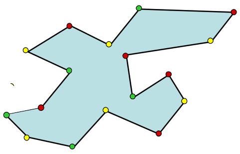
Proof by induction:

Base case (n=3):



Ind. hyp. (n≤m):

Ind. step (n=m+1): Polygon has an ear.

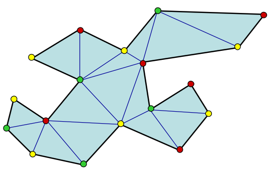


NATIONAL ICT AUSTRALIA

Main results

Theorem:

1. Every simple polygon can be triangulated.
2. Every simple polygon with n vertices can be guarded with $\lfloor n/3 \rfloor$ guards.



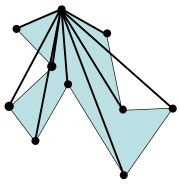
A triangulation exists but how can we compute it?

Triangulation algorithm 1

Theorem: Every polygon has a diagonal.

Testing a diagonal? O(n)

Algorithm 1:
    while P not triangulated do
        (x,y) := find_valid_diagonal(P)
        output (x,y)



Time complexity:
    #iterations = O(n)
    #diagonals = $O(n^2)$
    Test a diagonal = O(n)   ⇒ $O(n^4)$

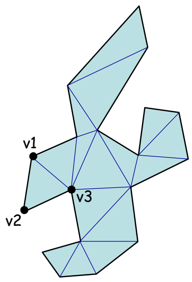
The imagination driving Australia's ICT future.

Triangulation algorithm 2

Theorem: Every polygon has at least two non-overlapping ears.

Algorithm 2:

while n > 3 do	n-3
locate a valid ear tip v2	$O(n^2)$
output diagonal (v1,v3)	O(1)
delete v2 from P	O(1)
	Total: $O(n^3)$

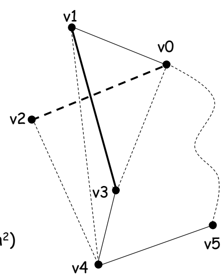


The imagination driving Australia's ICT future.

Triangulation algorithm 3

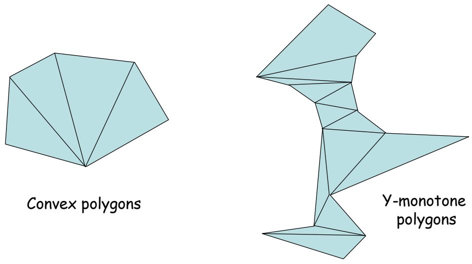
Algorithm 3:

compute all valid ears S	$O(n^2)$
while n > 3 do	n-3
locate a valid ear tip v2	O(1)
output diagonal (v1,v3)	O(1)
delete v2 from P	O(1)
delete (v0,v1,v2) from S	O(n)
delete (v2,v3,v4) from S	O(n)
check ear (v0,v1,v3)	O(n)
check ear (v1,v3,v4)	O(n)
	Total: $O(n^2)$



Triangulation algorithm 4

Observation: Some polygons are very easy to triangulate.



The imagination driving Australia's ICT future.

Polygon triangulation 4

Algorithm 4:

- Partition P into y-monotone pieces. O(n log n)
- Triangulate every y-monotone polygon O(n)

Theorem: Every simple polygon can be triangulated in O(n log n) time.

The imagination driving Australia's ICT future.

Polygon triangulation

- O(n log n) time [Garey, Johnson, Preparata & Tarjan'78]
- O(n loglog n) [Tarjan & van Wijk'88]
- O(n log* n) [Clarkson et al.'89]
- O(n) [Chazelle, Tarjan & van Wijk'91]
- O(n) randomised [Amato, Goodrich & Ramos'00]

Open problem: Is there a simple O(n)-time algorithm?