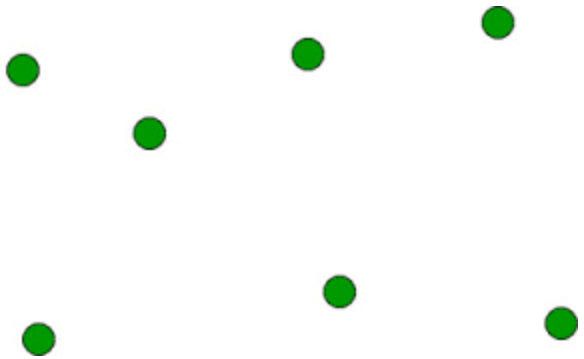
Proximity : Fundamental concepts and algorithms

Diameter of a point set

- **Given n points in a plane, find two that are the farthest apart.**



- **Set diameter problem**

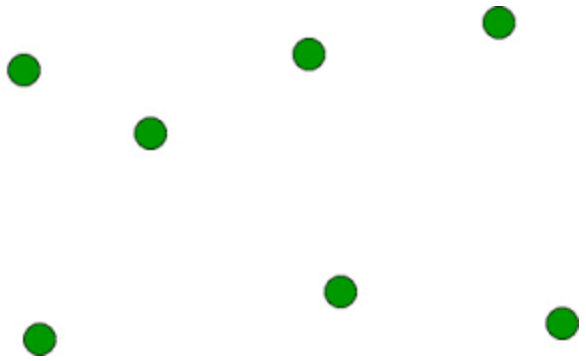
Set diameter problem

- What is a naïve algorithm for set diameter problem?
- How many pairs of points are there, if n is the total number of points ?
- (n(n-1))/2 pairs of points
- Compute the distance between each of the

  (n(n-1))/2 pairs of points and choose the largest
- What is the complexity?
- $O(n^2)$

Efficient solution

- Any other solution ?



- Convex hull

Convex hull again!!

- Theorem [*Hocking, Young 1961*]: The diameter of a set is equal to the diameter of its convex hull
- In the worst case all the n points will be on the hull
- We can solve the **Set diameter problem** by two steps:
- (1) Computing the convex hull of n points
- (2) Computing the diameter of a convex polygon
  - Shamos Algorithm

Diameter of a convex polygon

- Shamos algorithm
- Input : A convex hull (a convex polygon)
- Output: Diameter of the polygon
- Idea of the algorithm:
- Find all antipodal pairs of points from the convex polygon
- The **antipodal pair** that is at maximum distance apart achieves the diameter
- *What is an antipodal pair of points?*

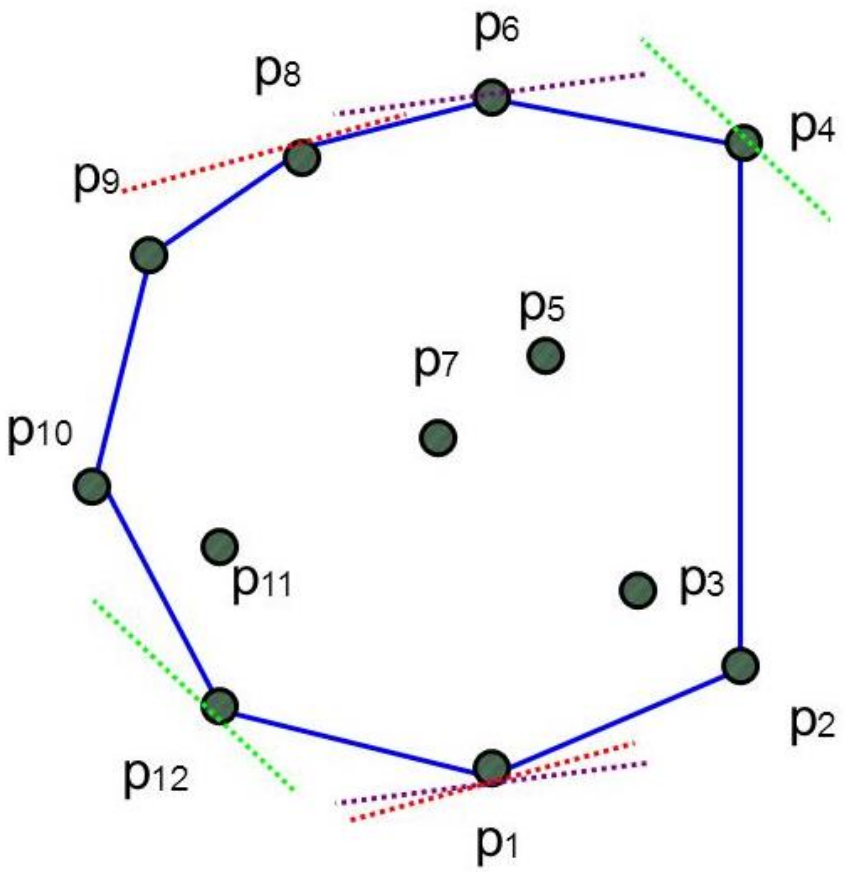
Antipodal pair of points

- An **antipodal pair of points** consists of two points such that:
- There exist two parallel lines, one through each point, and
- Every other point in the polygon lies between these two lines
- Exercise: Draw a convex polygon and an antipodal pair of points

Many antipodal pairs?

- Is it possible to have many antipodal pairs for a convex polygon?
- Exercise : Draw a convex polygon and many antipodal pairs of points

Antipodal pairs of points

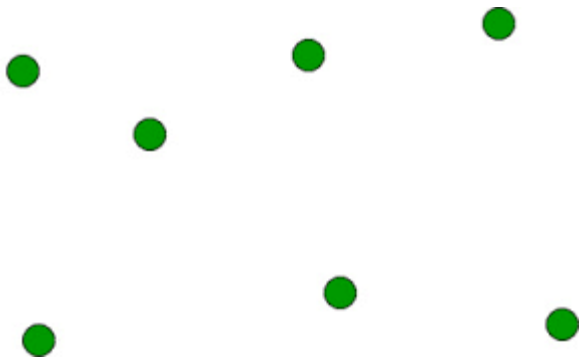


Exercise

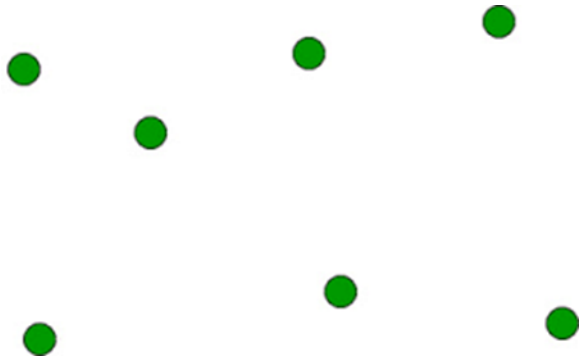
- Using the concept of antipodal pairs, the complexity to find the farthest pair of points can be reduced to O(n log n).
- Exercise for you : **Given n points in a plane, find two that are the farthest apart** can be solved in O(n logn)

Proximity

- We have discussed finding two farthest points of a set of points in a plane can be done in O(n log n) time
- **How do we find two closest points?**



Closest points



- Convex hull?
- The farthest points are necessarily hull vertices
- The closest points need not bear any relation to the convex hull
- To solve this we revive a classical mathematical object and turn it to an efficient computational structure – **Voronoi Diagram**

Voronoi Diagram

- The closest pair can be solved by:
- Using Voronoi diagram / The Locus approach

The Locus Approach to proximity problems

- A valuable **heuristic** for designing geometric algorithms is to :
- **Define Loci and try to organize them in a data structure**
- What is a heuristic?
- What is a loci ?

Heuristic in general [*Wikipedia*]

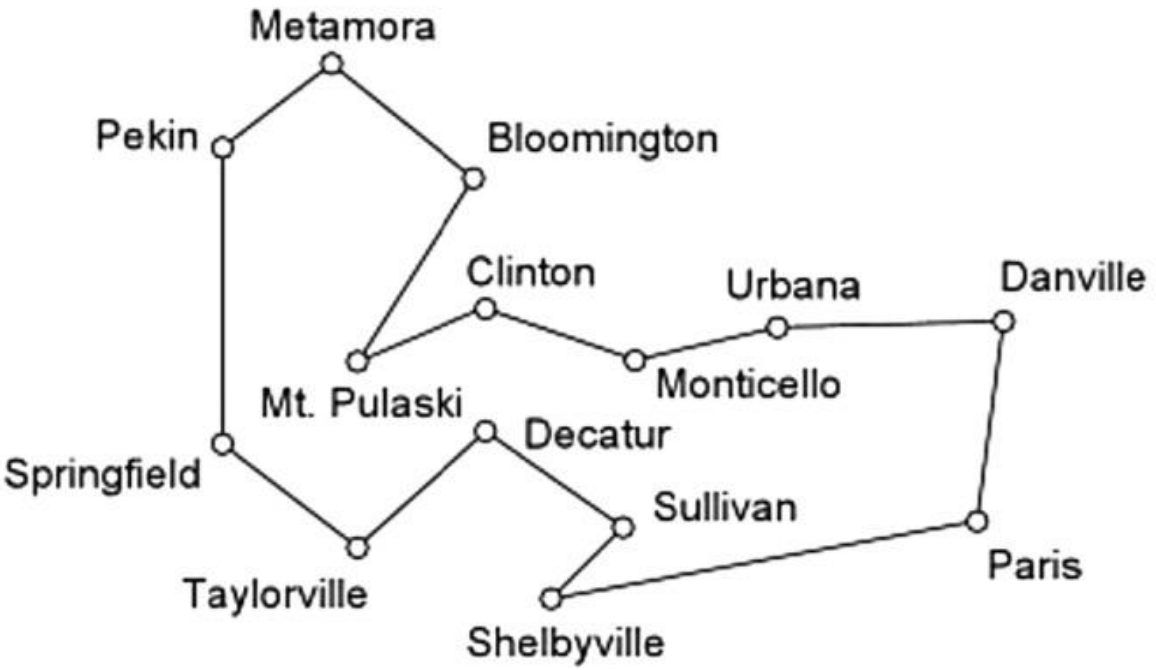
- A **heuristic** (*word meaning in Greek : I dicscover*) , is any approach to problem solving or self-discovery :
- that employs a practical method,
- not guaranteed to be optimal, perfect, logical, or rational,
- but it is sufficient for reaching an immediate goal,
- where finding an optimal solution is impossible or impractical, heuristic methods can be used to speed up the process of finding a satisfactory solution.

Heuristic in CS [*Wikipedia*]

- A **heuristic** is a technique designed for solving a problem :
- more quickly when classic methods are too slow
- or for finding an approximate solution when classic methods fail to find any exact solution.
- This is achieved by trading optimality, completeness, accuracy, or precision for speed.
- In a way, it can be considered as a shortcut.
- An example problem in CS where heuristic is used / an example of frequently used heuristic in CS?

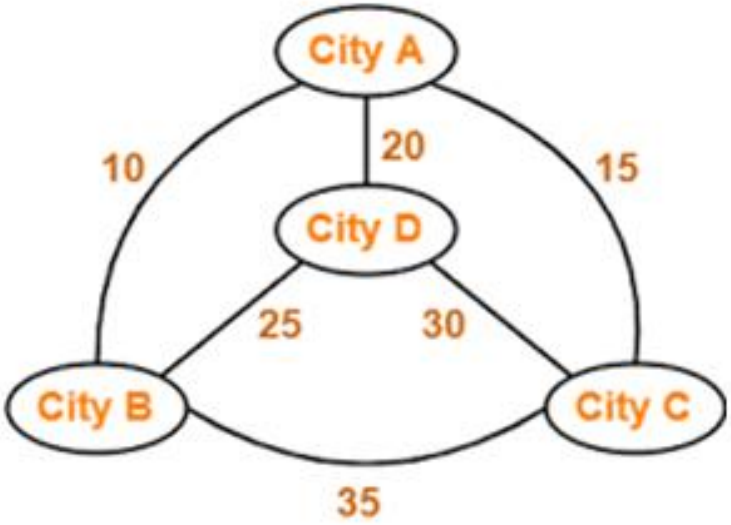
An example CS problem where heuristic is used [*Wikipedia*]

- Travelling Salesman Problem:



- Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city and returns to the original city?

Heuristic used



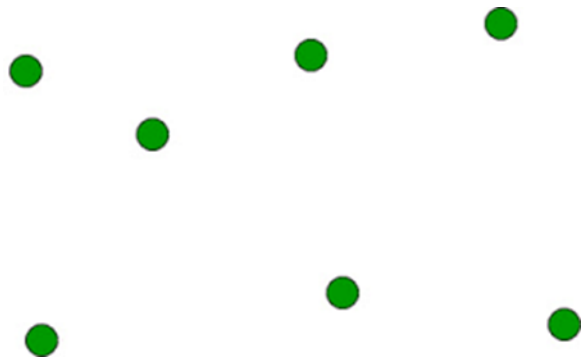
- The greedy algorithm heuristic / Nearest neighbor heuristic says to pick whatever is currently the best next step regardless of whether that prevents (or even makes impossible) good steps later.

The Locus Approach to proximity problems

- A valuable heuristic for designing geometric algorithms is to :
- **Define Loci and try to organize them in a data structure**
- What is a loci ?

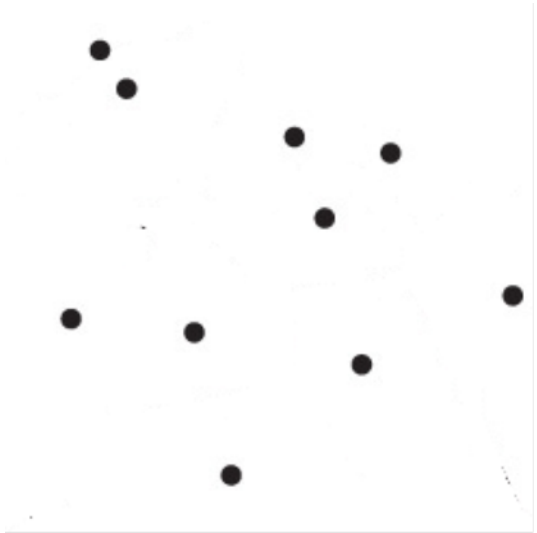
Locus / Loci [*Wikipedia*]

- In geometry, a **locus** (*Latin word for place or location)*
- is a set of all points (commonly, a line, a line segment, a curve or a surface),
- whose location satisfies or is determined by one or more specified conditions



Loci of proximity

- Given a set S of n sites/points in the plane, for each site / point p_i in S what is the locus of points (x,y) in the plane (consider all the infinitely many points in the plane) that are closer to p_i than to any other site/point of S ?



References

- F.P. Preparata & M.I. Shamos, *Computational Geometry An Introduction,* Springer International Edition, 1985
- J. O Rourke, *Computational Geometry in C*, 2/e, Cambridge University Press, 1998
- *https://cfbrasz.github.io/Voronoi.html* ---- Voronoi Diagram generator

THANK YOU