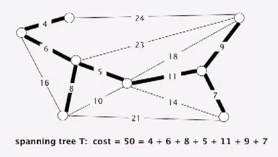
Minimum Spanning Trees

What is a mst?

Definition of graph in this case: An undirected graph G with positive edge weights (positive numbers associated with each edge); graph is connected

MAIN DEFINITION (spanning tree): A spanning tree of G is a subgraph T that is both a tree (connected and acyclic) and spanning (includes all of the vertices).

**Goal**: find a spanning tree (remember: connected and acyclic)

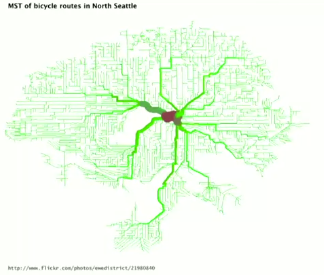


Brute force solution: try all spanning trees.

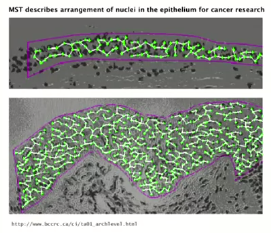
* Impractical
* Difficult to code

Spanning tree examples:

Network design (bike routes in N Seattle)



Medical image processing (nuclei arrangement in epithelium for cancer research)

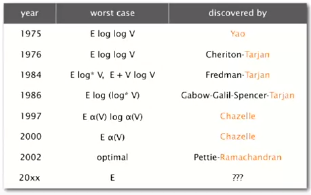


Many applications:

* Real-time face verification
* Image registration
* Find road networks in satellite/aerial imagery
* Network design
* Much much more\

MST Context

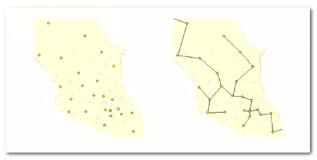
Does linear-time MST algorithm exist?



Short answer: We haven’t discovered a linear-time algorithm *worst-case*  
(Though in 1995, Karger-Klein-Tarjan discovered a linear-time *randomized* MST algorithm)

Applications

Euclidean MST: Given N points in the plane, find MST connecting them, where the distances between point pairs are their Eucliean distances.



Want subset of edges that connects all points that is minimal. Other algorithms suboptimal (E is N2)

1. **Brute force:** Compute ~ N2 / 2 distances and run Prim’s algorithm
2. **Ingenuity:** Exploit geometry and do it in ~ c N log N

Scientific Application: Clustering



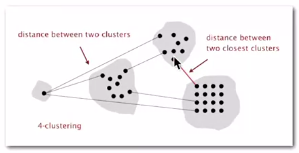
* k-clustering: divide a set of objects classified into k coherent groups
* Distance function: Numeric value specifying ‘closeness’ of two objects

Goal: Divide into clusters so that objects in different clusters are far apart

Applications:

* Routing in mobile ad hoc networks
* Document categorization for web search
* Similarity searching in medical image databases
* Skycat: cluster 109 sky objects into stars, quasars, galaxies

Single-link clustering



* **Single link:** distance between two clusters equals the distance between the two closest objects (one in each cluster)
* **Single-link clustering:** Given integer k, find k-clustering that maximizes the distance between two closest clusters.

‘Well-known’ algorithm for single-link clustering:

* Form V clusters of one object each
* Find the closest pair of objects such that each object is in a different cluster, and merge the two clusters
* Repeat until there are exactly k clusters
* This is Kruskal’s algorithm… just stop when there are k connected components!
* Alternatively: run Prim’s algorithm and delete k-1 max weight edges until you’re left with k clusters

Application: Dendrogram of cancers in human

