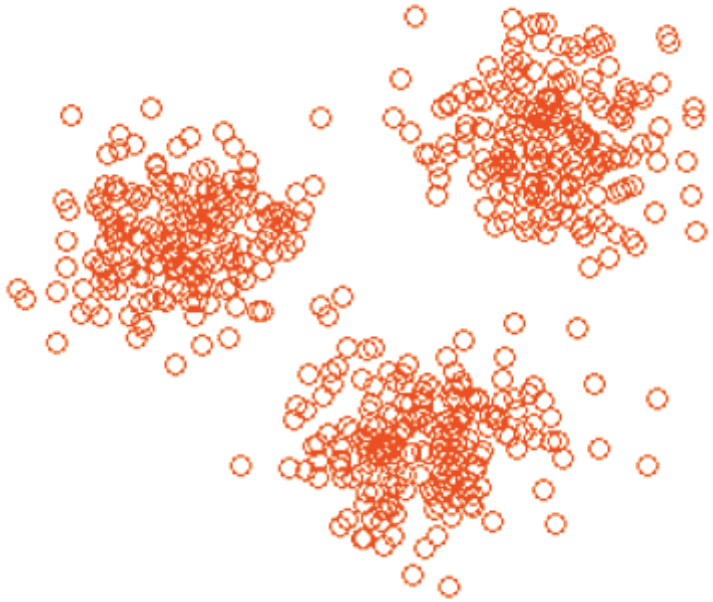
Graph Clustering

What is clustering?

- Finding patterns in data, or grouping similar groups of data-points together into *clusters*.
- Clustering algorithms for numeric data:
  - Lloyd's K-means, EM clustering, spectral clustering etc.



Examples of good clustering:

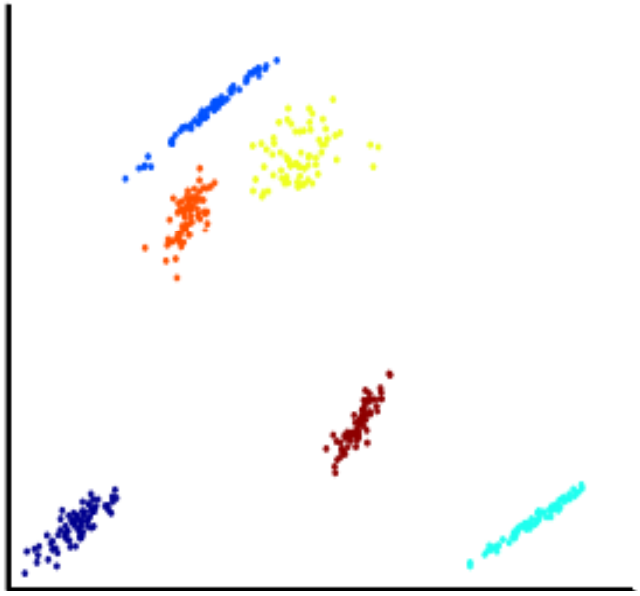
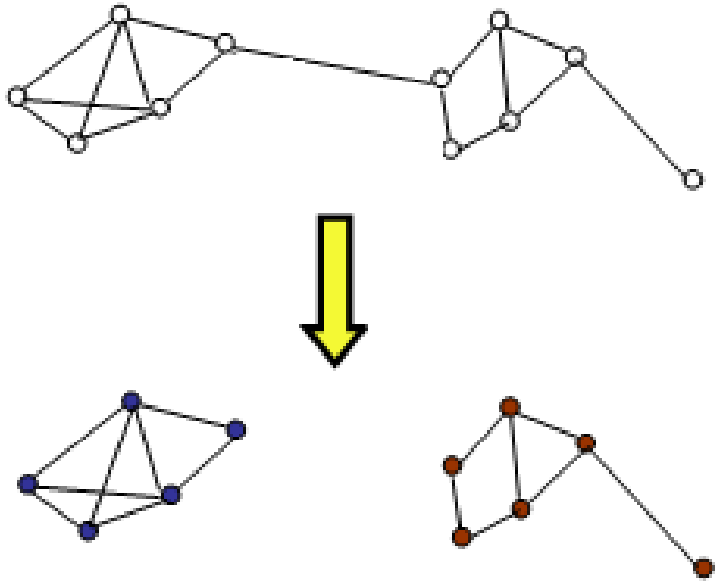


IMAGE SEGMENTATION

Graph Clustering:

- Graphical representation of data as undirected graphs.



GRAPH PARTITIONING!!

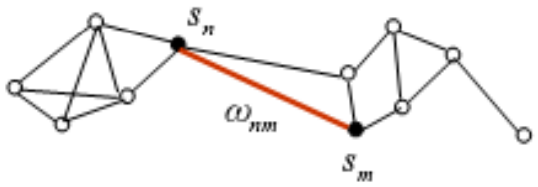
Graph clustering:

- Undirected graphs
- Clustering of vertices on basis of edge structure.
- Defining a graph cluster?
  - In its loosest sense, a graph cluster is a connected component.
  - In its strictest sense, it's a maximal clique of a graph.
- Many vertices *within* each cluster.
- Few edges *between* clusters.

Graph terminology:

pairwise affinity (similarity)

$$\omega_{nm} = e^{-\frac{|s_n - s_m|}{\sigma^2}}$$



node volume (degree)

$$D_n = \sum_{m=1}^{N} \omega_{nm}$$

volume of a set (cluster)

$$Vol(C) = \sum_{n \in C} D_n$$

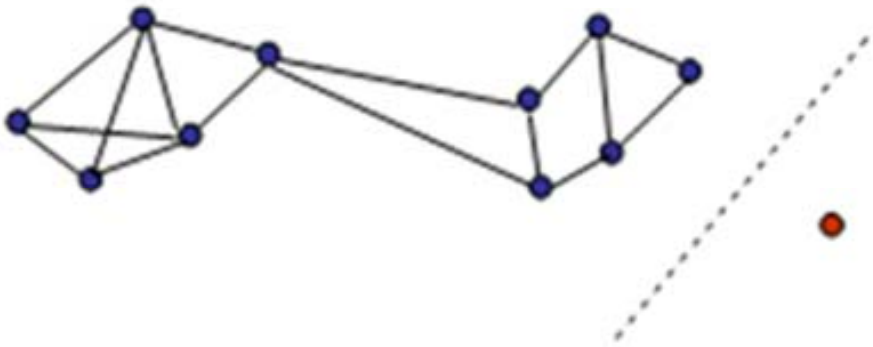
cut between 2 sets

$$Cut(C_1, C_2) = \sum_{n \in C_1} \sum_{m \in C_2} \omega_{nm}$$

Graph partitioning:

Minimal bipartition Cut:

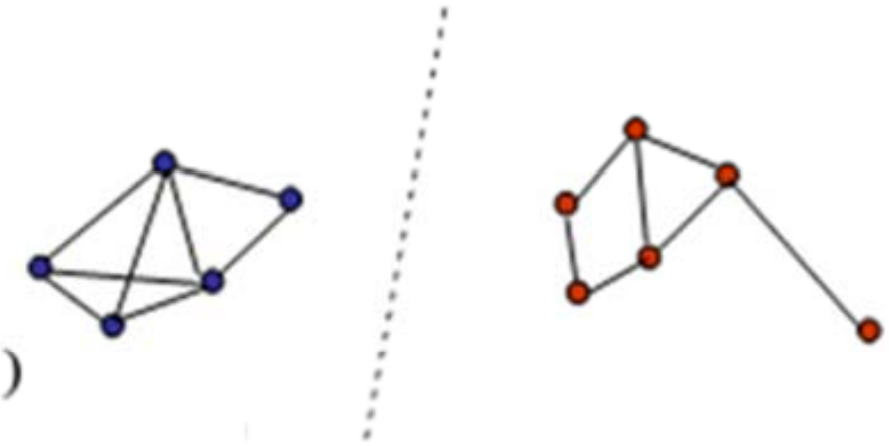
$$\min \; Cut(C_1, C_2)$$



Normalized miminal bipartition Cut:

$$= \min \frac{Cut(C_1, C_2)}{Vol(C_1)} + \frac{Cut(C_2, C_1)}{Vol(C_2)}$$

$$= \min \left(\frac{1}{Vol(C_1)} + \frac{1}{Vol(C_2)} \right) Cut(C_1, C_2)$$



Graph Partitioning:

- The optimization problem for normalized cuts is intractable (an NP hard problem).
- Hence we resort to spectral clustering and approximation algorithms.

More Graph notation:

Adjacency Matrix, A $a_{v,u}^{G} = \begin{cases} 1, & \text{if } \{v, u\} \in E, \\ 0, & \text{otherwise.} \end{cases}$

Degree Matrix $\mathbf{D} = \begin{pmatrix} \deg(v_1) & 0 & 0 & \dots & 0 & 0 \\ 0 & \deg(v_2) & 0 & \dots & 0 & 0 \\ 0 & 0 & \deg(v_3) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \deg(v_{n-1}) & 0 \\ 0 & 0 & 0 & \dots & 0 & \deg(v_n) \end{pmatrix}$

The properties of the *Laplacian of a graph* are found to be more interesting for the characterizatiojnof a grpah than the adjacency matrix. The unnormalized Graph Laplacian is defined as

$$L = D - W$$

Properties of the Laplacian:

1. For every vector $f \in \mathbb{R}^n$

$$f'Lf = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$$

2. L is symmetric and positive definite.
3. 0 is an eigenvalue of the Laplacian, with the constant vector as a corresponding eigenvector.
4. L has *n* non-negative eigenvalues.

$$0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$$

Number of Components:

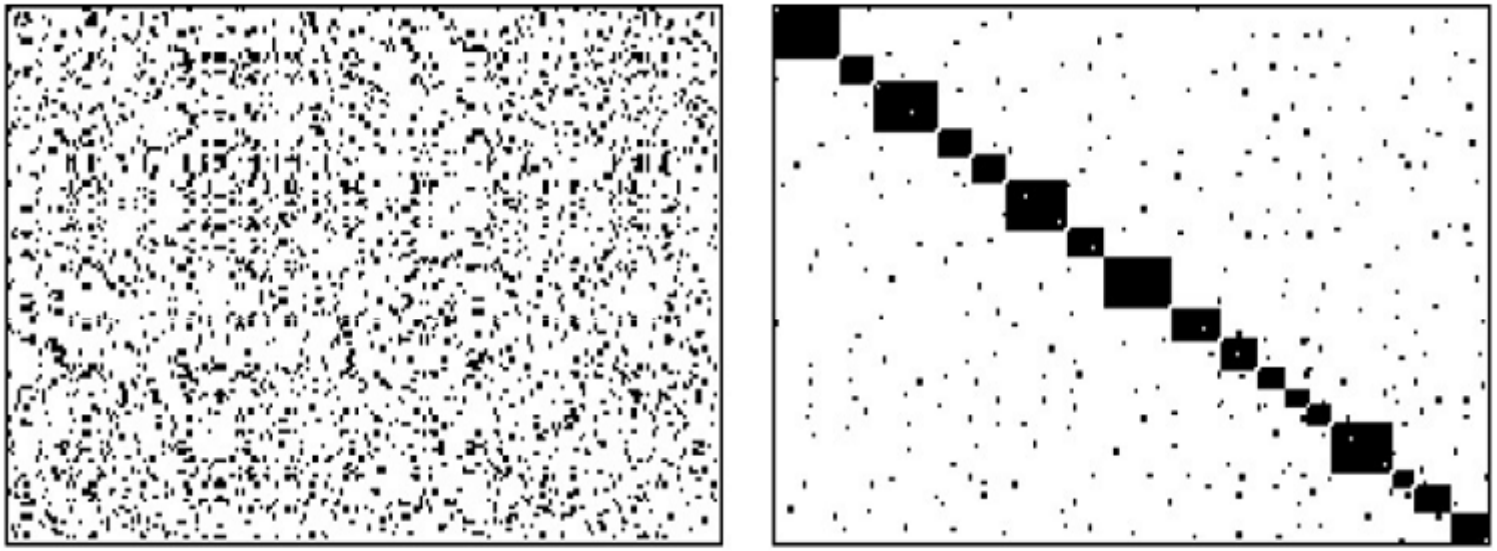
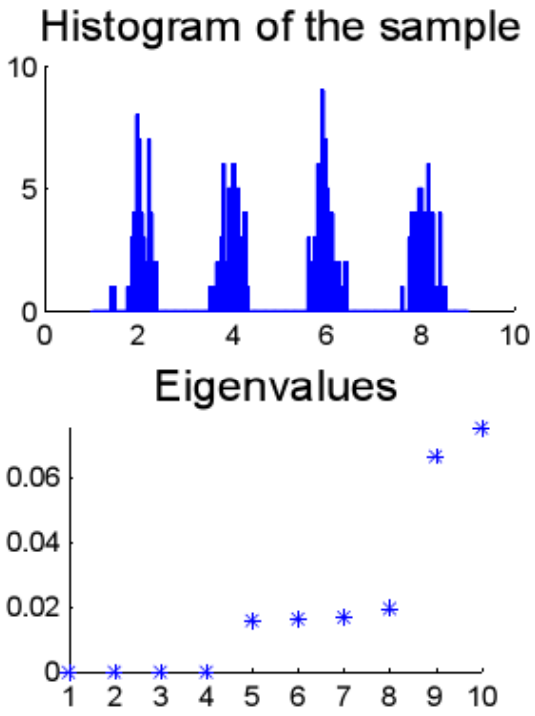
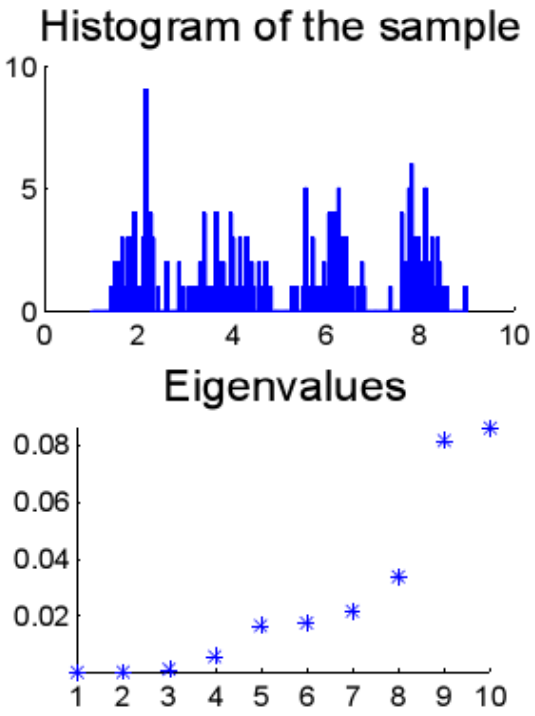


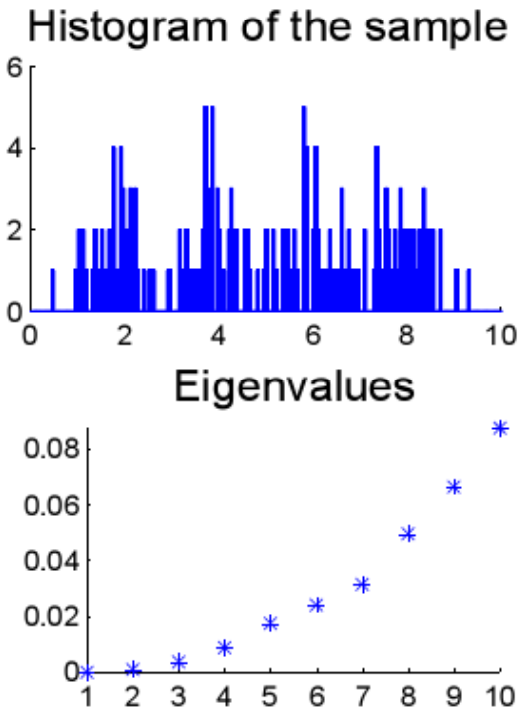
Fig. 1 – The adjacency matrix of a 210-vertex graph with 1505 edges composed of 17 dense clusters. On the left, the vertices are ordered randomly and the graph structure can hardly be observed. On the right, the vertex ordering is by cluster and the 17-cluster structure is evident. Each black dot corresponds to an element of the adjacency matrix that has the value one, the white areas correspond to elements with the value zero.

Graph spectra:

- The multiplicity of the eigenvalue 0 gives the number of connected components in the graph.





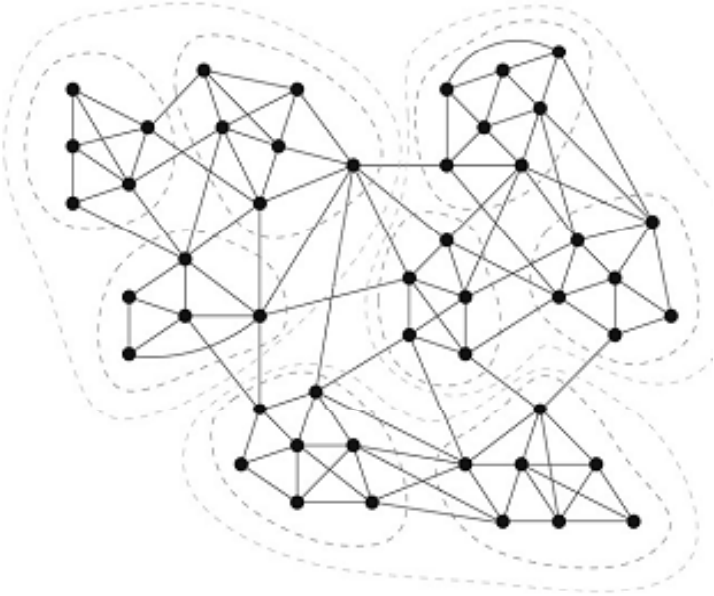


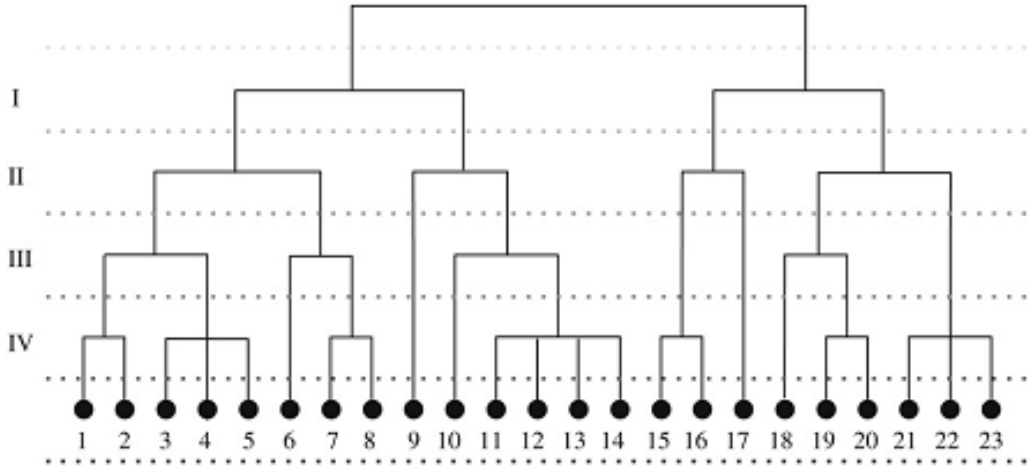
Graph Generation models:

- Uniform random model
  - All edges equiprobable
  - Poissonian degree distribution
  - No cluster structure.
- Planted partition model
  - *l* partitions of vertex set
  - Edge-probabilities *p* and *q*.
- Caveman graphs, RMAT generation etc.
- Fuzzy graphs??

General clustering paradigms:

- Hierarchical clustering VS flat clustering.
- Hierarchical:
  - Top down
  - Bottom up





Overview:

- Cut based methods:
  - Become NP hard with introduction of size constraints.
  - Approximation algorithms minimizing graph conductance.
- Maximum flow
  - Using results by Golberg and Tarjan
  - Reasonable for small graphs.
- Graph Spectrum based:
  - Stable perturbation analysis
  - Good even when graph is not exactly block diagonal.
  - Typically, second smallest eigenvalue is taken as graph characterstic.
  - Spectrum of graph transition matrix for blind walk.

Overview:

- Could experiment with properties of different Laplacians.
- Typically outperforms k-means and other traditional clustering algorithms.
- Computationally unfeasible for large graphs.
- Roundabouts?

Voltage-potential view: ☺

- Related to 'betweenness' of edges.

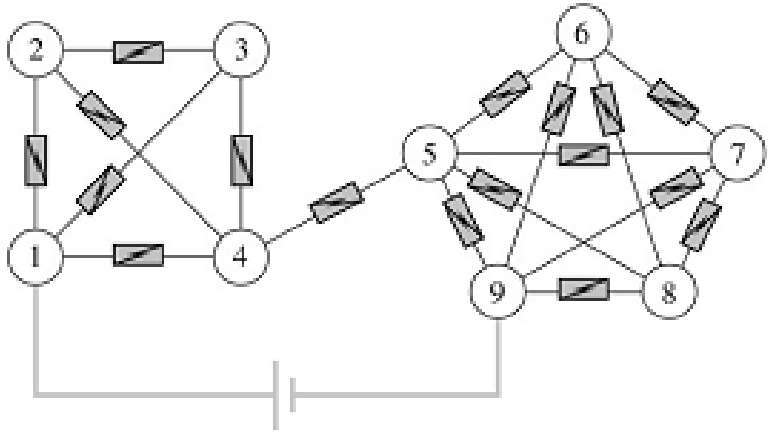
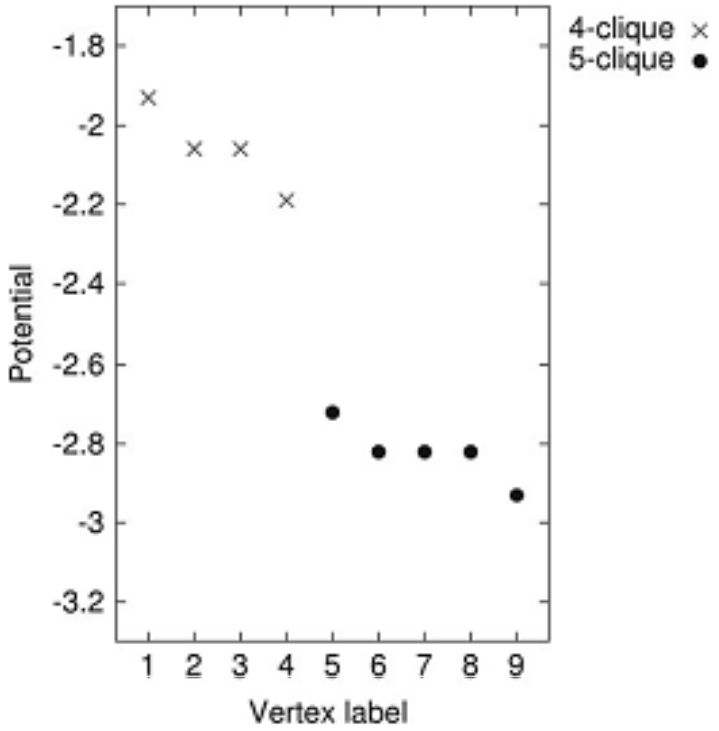


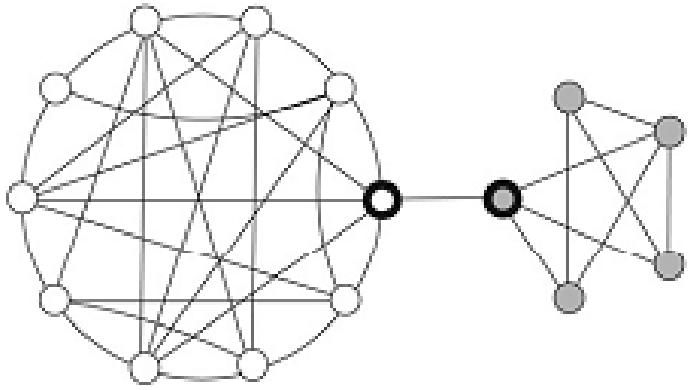
Fig. 6 – A graph transformed into an electric circuit: a unit resistor has been placed on each edge, and a battery connects the vertices labelled one and nine (battery connection drawn in grey).



- Not stable to placement of rando m sources and sinks.

Markov Random walks:

- Vertices in same cluster are quickly reachable.
- A random walk in one of th clusters is likely to remain for a long time.



- The Perron-Frobenius theorem ensures that the largest eigenvalue associated with a transition matrix is always 1. (relation with Graph Laplacian).
- Component of second eigenvector vector of the transition matrix serves as a measure of absorption time.

Thank you.