CS 594: Representations in Algorithm Design	Spring 2022

Lecture 5: 25th January 2021

Lecturer: Xiaorui Sun — *Scribe: Kumar Chandrasekhar*

1 Recap of Last Lecture

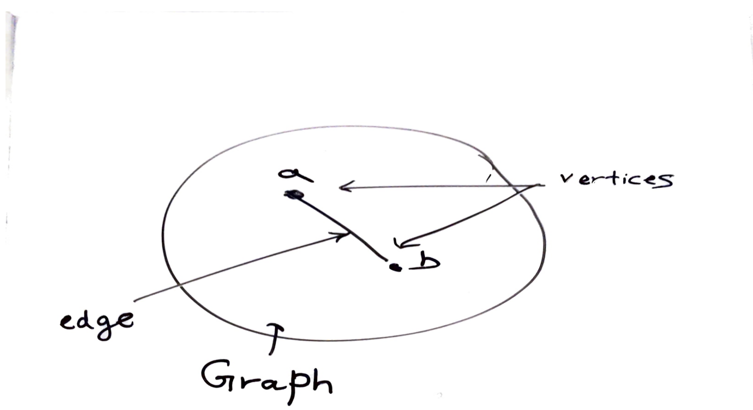

Figure 1: Graph Diagram

- The standard way to compute distance between two vertices in a graph is Djkstra's algorithm.
- The time complexity of the algorithm is O(m) where m is the number of edges in a graph.
- Our aim is to see if we can compute the distance between two points faster.
- We have observed that for a given tree T, which is rooted, the distance between two vertices 'u' and 'v' given as: $d(u,v) \longrightarrow O(d(u,v))$.

- This becomes possible in a tree, whereas the same is not the case in graph.
- Hence, we introduce **Low-Stretch Spanning Tree** through which we attempt to reduce the computation time of distance between two vertices in a graph.

1.1 Low Stretch Spanning Tree

- Our goal is that, given a graph G, we try to form a spanning tree such that $d_{Tree}(u, v) \approx d_{graph}(u, v)$
- Consider a graph G = (V,E), where V represents the set of vertices and E represents the set of edges. Then, for an arbitrary edge $[e \in E]$, we define the term Stretch as follows:
  $Stretch_T(entiregraph) = \dfrac{distance_{T,W}(u,v)}{W(e)_{E,G}}$
  Stretch of entire graph is given as:

  $Stretch_T(entiregraph) = \sum_{e \in E(G)} Stretch_T(e)$
- Stretch can be considered in general sense as a measure to minimize distance between two vertices u and v in a graph.
  Though, having said that, we cannot compare two spanning trees T_1 and T_2 using $Stretch_T$, but, in general, it is the average representation of distance between vertices.

- So, what is the goal of using a low stretch spanning tree?

  Goal: Given a graph G with n vertices and m edges, find a spanning tree T, such that:
  $m \leq Stretch_T \leq n \cdot m$
  Lemma: For a given graph G, there exists a spanning tree T such that:
  $Stretch_T \geq \Omega(m \cdot log(n))$

2 This Lecture

In this lecture, we are going to look into an algorithm that has a stretch of spanning tree with complextiy O($m \cdot 2^{\sqrt{logn.loglogn}}$).

The above mentioned complexity is worse than $O(m \cdot log^2 n)$, but better than, $O(m \cdot n^{1.0001})$

Now, let us look into an example, where we take a grid-graph and try to compute low stretch spanning tree for it. The grid graph has n vertices and hence, $\sqrt{n}$ vertices in rows and $\sqrt{n}$ vertices in columns.

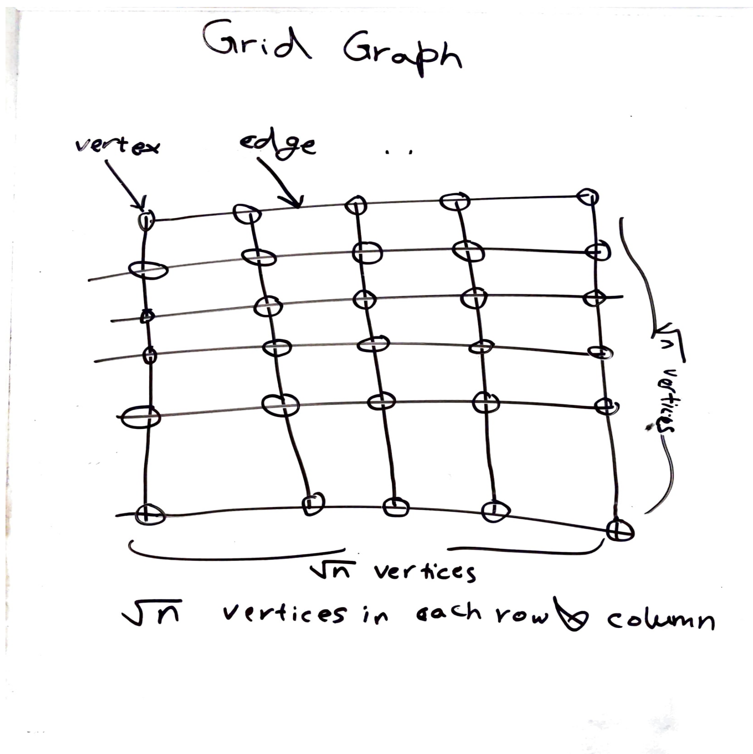

Figure 2: Grid Graph

- The first spanning tree that we would consider is the single path spanning tree. Let the vertex in top left corner of grid graph be a and the bottom right corner be b. Then, for a single path spanning tree, the distance(a,b) = n which is pretty bad.
- Let us define a parameter called diameter for a graph G, with set of vertices V: Diameter $i = distance_{u,v \in V}(u, v)$. Diameter is a measure to approximate a graph with a spanning tree. For a grid graph, Diameter $i = O(\sqrt{n})$.

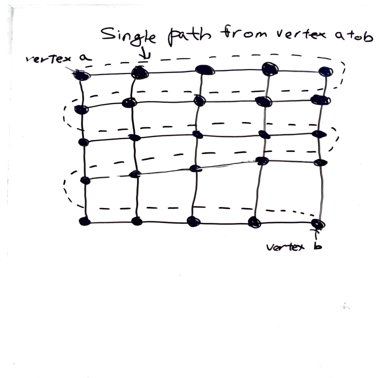

Figure 3: Single path spanning tree

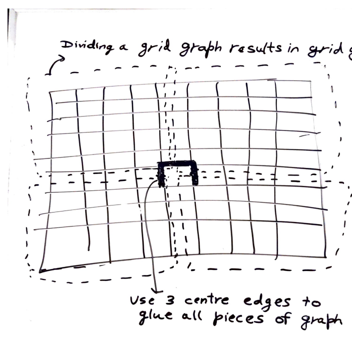

Figure 4: Gluing 4 pieces of divided grid graph

If you divide a grid graph, you will still end up with a grid graph. Now, if we divide a grid graph into 4 pieces and we have a spanning tree for each of the four pieces, how do we glue them together? **Answer:** We use the centre 3 edges of the grid graph.

After dividing a grid graph into 4 sub-grids, the total number of edges missing, as compared to the original graph is $2\sqrt{n}$.
If we minimize the number of edges missing, we can minimize the damage to the representation of the original grid graph.

Now, given a grid graph G, our goal is to find a spanning tree T such that $diameter(T) \approx diameter(G)$. Given that there are n vertices in graph G,

$Stretch_G(n) = 4 * Stretch(\frac{n}{4}) + O(\sqrt{n}) \cdot O(\sqrt{n}) = O(n \cdot logn)$ Here, we perform recursion, each time dividing any sub-grid-graph into 4 sub-grid graphs.

2.1 Low Diameter Decomposition(LDD)

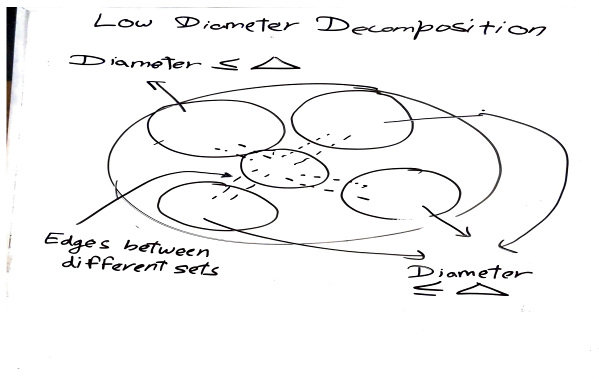

Figure 5: Low Diameter Decomposition

Goal: To partition the graph vertices into few sets $S_1, S_2, ...S_k$, where k is an arbitrary number. There are 2 properties of low diameter decomposition:

1. Within each set s_i, $diameter(G[S_i]) \leq \Delta$, where Δ is a user-given parameter.
2. Number of edges crossing different sets is small compared to m, which is the total number of edges in the graph G.

Given a large graph, when we perform LDD on it, we get sub-graphs such that the diameter of sub-graph g_i, $diameter(g_i \leq \Delta)$.
The number of edges between different sub-graphs is not large, when compared to m.
If Δ is small, then the edges

For a line graph, given a Δ, the number of edges crossing sets is given as:

$$E_{cs} \approx \frac{m}{\Delta}$$

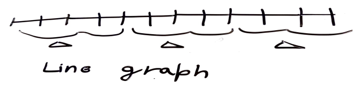

Figure 6: Graph Diagram

Lemma: We can perform LDD such that, for a given Δ:

- Number of crossing edges is proportional to $O(\frac{mlogn}{\Delta})$

2.1.1 Algorithm to construct LDD

- Select arbitrary vertex X in graph G.
- Let X grow a ball B.
  $B(X, i) = V : dist(X, V) \leq i$ when i=0, $BX, 0 = X$
  when i=1 $B(X, 1) = X \cup neighboursofX$
  when i=2 $B(X, 2) = B(X, 1) \cup neighboursofB(X, i-1)$
  Generalizing we get,

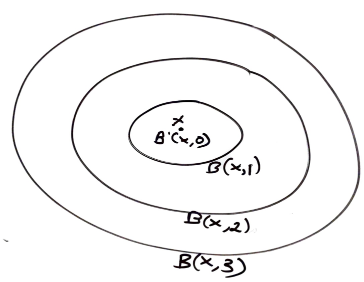

Figure 7: Ball Diagram

$B(X, i) = B(X, i-1) \cup neighboursofB(X, i-1)$

- If we think in terms of BFS(Breadth first search), the root vertex is B(X,0), the vertices in the first layer including the root vertex is B(X,1), B(X,2) is vertices in second layer plus B(X,1) ans so on....
- If we think in context of BFS:

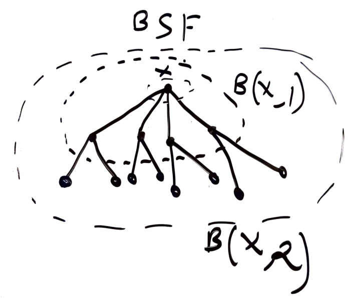

Figure 8: Ball Formation for BSF

- This process of ball formation goes on until the number of edges in a ball meet the below inequality:
  $|E(B(X, i+1))| \leq (1 + \frac{logn}{\Delta}) \cdot |E(B(X, i))|$
- Remove all vertices and edges from graph G. Let B(X,i) be a set.
- Repeat on the remaining graph.

Note: Number of red edges $\leq \frac{logn}{\Delta} \cdot (blueedges)$.

2.1.2 Properties of LDD

Let us look into two properties of LDD along with their proofs as given below:

1. LDD gives a set with a diameter $\leq \Delta$

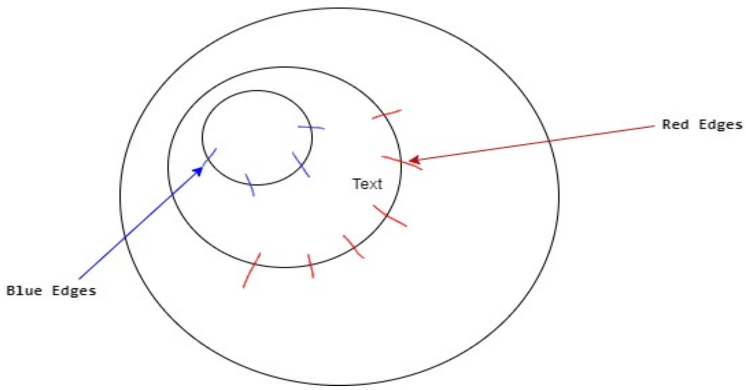

Figure 9: Red and Blue edges in the ball diagram

2. Number of edges crossing sets $\leq \frac{mlogn}{\Delta})$
   If we extract a set with E(B(X,i)) within the set, the cost will be proportional to E(B(X,i),B(X,i+1 (X,i))) $\leq |\frac{logn}{\Delta}| \cdot E(B(X,i))$

Now, let us look at proof for second property mentioned above.
We know,
$|E(B(X,0))| = 0$
$|E(B(X,1))| = 1$
$|E(B(X,2))| > (1 + \frac{logn}{\Delta})$
$|E(B(X,3))| > (1 + \frac{logn}{\Delta}) \cdot |E(B(X,2))| = (1 + \frac{logn}{\Delta})^2$
$|E(B(X,i))| \geq (1 + \frac{logn}{\Delta})^{i-1}$
$i = \Delta, E(B(X,\Delta)) \geq (1 + \frac{logn}{\Delta})^{\frac{\Delta}{logn} \cdot logn}$
The above quantity is of the form $(1 + \frac{1}{k})^k \implies e$, when $k \longrightarrow \infty$

3 Summary of Today's Lecture

- Algorithm to do Low Diameter Decomposition
- How do you construct Low Stretch Spanning Tree with Stretch equals $O(m\sqrt{n})$.

- Based on LDD, $\Delta = \sqrt{n}$