Visibility Graphs of Staircase Polygons: Algorithm for Building Staircase Polygons from Slope-Ranking Balanced Tableaux

James Abello, Yulia Alexandr

Abstract—There exists a relationship between staircase polygons, persistent graphs, and balanced tableaux [1]. While many aspects of this relationship had been rigorously studied in [1], the problem of recovering a simple staircase polygon whose visibility graph is isomorphic to the skeleton of a given slope-ranking balanced tableau remained open and was previously known to be PSPACE. In this paper, we demonstrate a deterministic polynomial-time algorithm for constructing a staircase polygon with desired properties and prove that certain conditions required by the algorithm hold for any balanced tableau.

1 Introduction

First, this paper follows the definitions and concepts introduced in [1], as this work adds on to what has already been discovered by J. Abello, O. Egecioglu and K. Kumar.

It has been shown that visibility graphs of staircase polygons are persistent. Moreover, there has been discovered a polynomial-time algorithm that recovers a representative maximal chain in the weak Bruhat order from some given persistent graph [1]. The problem of recovering a simple staircase polygon whose visibility graph is isomorphic to the skeleton of a given slope-ranking balanced tableau T_n was introduced in [1] and had been open since then.

- *Dr. James Abello is at DIMACS, Rutgers University.*
- *Yulia Alexandr is at Wesleyan University.*

The algorithm we discuss in this paper starts building from the middle vertex (or one of the two middle vertices) of a balanced tableau, thus taking advantage of the unboundedness of construction in both directions, and iteratively constructs visibility regions where we place new vertices. We also prove that the algorithm is deterministic and works for any balanced tableau by showing that the visibility regions we define are never empty as long as we preserve relative slope ranks of farthest seen vertices at every stage of construction and that it is always possible to preserve the needed slope ranks.

2 Definitions

We will denote a slope-ranking balanced tableau of n points in the plane with $\binom{n}{2}$ entries as T_n. We will denote the skeleton of T_n by $S(T_n)$. To refer to a particular entry in the column i and row j, we will write T_{ij} for the entry in tableau and $S(T_n)_{ij}$ for the entry in its skeleton.

We assume that the reader is familiar with the fact that it suffices to look at the core of a staircase polygon, as it contains example-specific visibility information. Thus the algorithm we discuss builds the path $v_1v_2...v_n$ (where $v_1, v_2, ..., v_n$ are the points where the staircase path changes from a vertical segment

to a horizontal one), which automatically implies a particular staircase polygon. We refer to such $v_1v_2...v_n$ as a **visibility path**.

When comparing slopes, we say that the slope of some line m is larger than the slope of another line k if m is ranked higher than k in a slope-ranking tableau (i.e. if the slope of m is more negative than the slope of k).

We define a **concave-concave path** to be two concave paths joined by a convex vertex. Similarly, we define a **concave-convex path** to be a concave and a convex paths joined together.

We define a **visibility region** to be a region in the plane defined by lines formed by visible vertices such that if we place a vertex p anywhere in the region, its visibility (as defined by $S(T_n)$) is satisfied with respect to the other (already placed) vertices in the path. The lines that form a visibility region at each stage of construction are defined as follows. Suppose we want to construct a region for some vertex p. For each already placed vertex v, we look at the farthest vertex it sees before p, call it w. Then we draw the line defined by v and w (or extend an already existing edge). If p sees v (i.e. $S(T_n)_{pw} \equiv 1$), then the region is defined on the left of vw, and on the right if otherwise. The intersections of such regions $\forall v$ gives us the visibility region for p. One can see that, by the nature of staircase polygon paths, this procedure makes sure that the visibility of p is satisfied, as w would be the vertex blocking p from seeing v if $S(T_n)_{pw} \equiv 0$ and wouldn't prevent p from seeing v if $S(T_n)_{pw} \equiv 1$, since it is the farthest vertex v sees before p.

3 The Algorithm

Input: A slope-ranking tableau T_n.

Output: A staircase polygon whose visibility graph is isomorphic to the skeleton of the tableau, $S(T_n)$.

We begin building a visibility path from the middle vertex (or one of the two middle vertices) of the tableau, v. Then we place its visibility path neighbors $(v-1)$ and $(v+1)$. If $S(T_n)_{(v-1)(v+1)} \equiv 0$, then $slope((v-1), v) > slope(v, (v+1))$ and hence the three-vertex subpath has a concave shape, since $((v-1), (v+1))$ is in the exterior of the visibility path and hence not in the visibility graph of the polygon. If $S(T_n)_{(v-1)(v+1)} \equiv 1$, then $slope((v-1), v) < slope(v, (v+1))$ and hence the three-vertex subpath has a convex shape. We record the values of the slopes of the edges $((v-1), v)$ and $(v, (v+1))$. We also record the value of $slope((v-1), (v+1))$ if $(v-1)$ sees $(v+1)$. We then store the recorded slopes according to their ranks.

Moving forward, we add vertices iteratively two at a time, one at each end of the existing path. (If there is only one vertex left, we simply construct its visibility region separately). We construct the visibility region for each new vertex using the lines between the farthest visible vertices we have already placed at that point in time. As we do this, we preserve slope ranks of farthest seen vertices of each vertex at every stage of construction relative to the slopes we've already recorded up to this iteration. We place each new vertex in its region such that the slope ranks of the vertex and the already placed vertices it sees are preserved and record the preserved slopes. By the way visibility regions are defined, any point in its non-empty visibility region will satisfy its required visibility. Thus, this algorithm will recover a visibility path for any slope-ranking balanced tableau as long as there are no empty visibility regions and as long as it is always possible to preserve relative slope ranks as descried above.

4 Empty Regions

Clearly, the algorithm described in the previous section breaks if a visibility region happens to be empty. However, in this section, we prove that empty visibility regions are not possible as long as we preserve relative slope ranks between each vertex and the farthest vertex it sees at each stage of construction. We first prove this for two special cases and then present two independent proofs for the general case.

Note that for the special cases, we modify our algorithm to build a visibility path starting with the first vertex, adding consecutive vertices one at a time.

4.1 *Impossibility of Empty Regions for Special Cases*

Lemma 4.1. *For a concave-concave path on n vertices constructed iteratively from vertex 1 to n, there always exists a non-empty visibility region for every vertex v as long as slope ranks of farthest seen vertices of each vertex preceding v are preserved at every stage of construction.*

Proof. Suppose we have a tableau whose skeleton is isomorphic to the visibility graph of some concave-concave path. Let the vertex numbered k be the convex vertex separating two concave subpaths of that path. It is easy to construct the path for the first k vertices.

By definition of concave-concave paths, it suffices to look at the rectangular subgraph of the skeleton (all entries below and including the convex vertex row). Since the visibility graph of the given tableau is persistent, it must be the case that the row numbered $(k+1)$ has a block of consecutive ones, and all the entries to the left of this block have to be zeros. Similarly, by the persistence property, each of the next rows have blocks of ones that, if overlap for some one-entries from the above row, may extend to the left of the block above or lose some one entries in the right end. If the one-blocks of two consecutive rows do not have any overlaps, it must be the case that the second row consists of only zero entries. This means that for the vertices $(k+1)$ through n, there are paths of consecutive visible vertices among vertices 1 through k.

Suppose we have built a path 1 to k. Also, suppose $(k+1)$ sees all the vertices from $(k-m)$ to $(k-1)$; $1 < m$. By construction, the edges $((k-m-1),(k-m))$ and $((k-m),(k-m+1))$ determine the visibility region for $(k+1)$: clearly $(k+1)$ must be placed in-between the extensions of the above edges. Placing $(k+1)$ in the region preserving slopes and connecting k and $(k+1)$ extends the path.

Now we want to show that the region for $k+2$ is never empty. After we place the point $(k+1)$ in its region, we will draw the line p through this point parallel to the edge $((k-m-1),(k-m))$ (Figure 1). $\forall v \in \{(k-m),(k-m+1),...,(k-1)\}$, if v is to the left of p, $(v,(k+1))$ does not intersect the extension of $((k-m-1),(k-m))$ below; if v is to the right of p, then it does. The vertices that do not intersect the extension of $((k-m-1),(k-m))$ below may potentially create an empty visibility region for the next point in the path, if the new point sees $(k-m-1)$ and does not see one or more of points to the left of p.

Case 1: If $(k+2)$ does not see some of the vertices on the right of p and sees some of them, then it must see all the vertices on the left, since the block of ones has to continue to the left, by persistence property.

Case 2: If $(k+2)$ does not see any of the vertices on the right of p.

Case 2.1: If $(k+2)$ also does not see any of the vertices on the left, by persistence property, the $(k-2)$ row of the tableau must have all zero entries, and we can always find the region for a vertex that does not see any of the other vertices.

Case 2.2: If $(k+2)$ sees all the vertices to the left of p, there is no contradiction with the extension of $((k-m-1),(k-m))$, since the region is defined on the left of all lines.

Case 2.3: If $(k+2)$ does not see some of the vertices on the left of p, it has to be a subcollection of consecutive vertices on the right end. If the region is defined on the right of the lines connecting those vertices and $(k+1)$, extensions of the edges between $(k+1)$ and the points on the left of p visible to $(k+2)$ will cross those lines and, since the region is defined to the left of those extensions, the region will not be empty.

Case 3: If $(k+2)$ sees all the vertices on the right of p, it must see all the vertices on the left of p, and there is no problem.

We shall iteratively do the same with the remaining vertices in $\{(k+1),(k+2)...,n\}$, since by the nature of concave-concave paths, none of the vertices in this set see each other (except for the pairs of adjacent vertices in the path) and all of them have the same properties as $(k+1)$ and $(k+2)$.

□

Lemma 4.2. *For a concave-convex path on n vertices constructed iteratively from vertex 1 to n,*

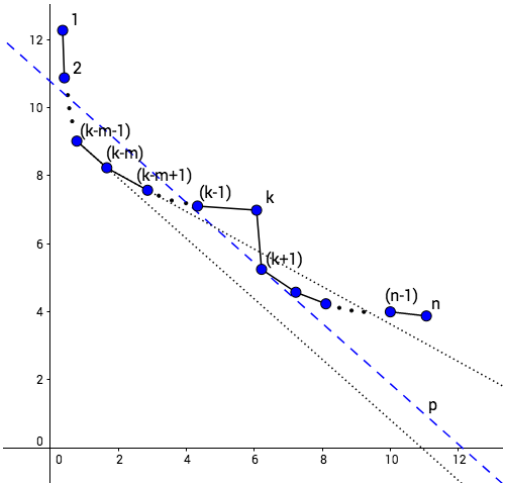

Fig. 1: Concave-Concave Path

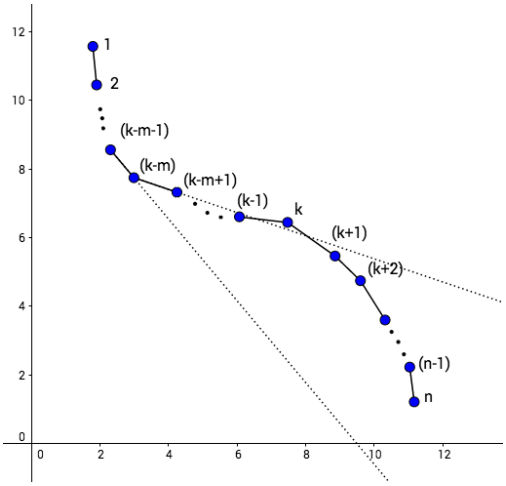

Fig. 2: Concave-Convex Path

there always exists a non-empty visibility region for every vertex v as long as slope ranks of farthest seen vertices of each vertex preceding v are preserved at every stage of construction.

Proof. Let k be the first convex vertex separating the concave and convex subpaths.

The first observation is that, by persistence property, every new vertex after the first convex vertex sees at least everything the previous vertex sees. Moreover, $(k+1)$ always sees $(k-1)$, otherwise k would not be the first convex vertex. Hence, every row has a block of consecutive ones starting at the right end of the tableau rectangle described in the proof before. With every new row, the block can only extend to the left, so every previous vertex sees only a subset (not necessarily proper) of what the next vertex sees.

So, the region for $(k+1)$ is formed in the same way as in the concave-concave case. It is important to notice that since the second subpath is convex, every new vertex $(k+l), l > 0$ sees a collection of consecutive vertices right before itself, i.e. up to $(k+l-1)$.

Then the only way we could potentially get an empty region for a new point is if $(k+l)$ sees some v_1 and doesn't see some v_2 in the first subpath, $v_1 < v_2$. Yet, this is not possible, since by construction $(k+l)$ sees a collection of consecutive entries up to itself.

□

By symmetry with the above cases, empty visibility regions are also not possible for convex-convex and convex-concave paths.

4.2 *Impossibility of Empty Regions for the General Case*

Lemma 4.3. *As we construct a path from the middle of the tableau, iteratively adding one vertex at each side while preserving slope ranks of edges between a vertex and its farthest seen vertex at every stage, empty visibility regions are not possible for any new point.*

Proof. By Induction on k (the number of lines constructing the region of the n^{th} vertex).

Base Case:

Suppose we have a polygon $P[2, n-1]$ that has a non-empty visibility region for 1. If it has a non-empty region for n, we're done. Suppose it doesn't and the visibility region for n is empty in this polygon. This means the intersection of the regions defined by lines that produce it is empty. For the base case, assume there are two such lines that contradict each other.

WLOG, assume that the two contradictory lines are the lines a (produced by two vertices p_1 and p_2) and b (produced by p_3 and p_4) such that n sees p_3, so the region is on the left of b and n doesn't see p_1, meaning the region is on the right of a. Since the region for n is assumed to be empty, $slope(b) > slope(a)$ and

the lines cross before $(n-1)$. Since the lines are extensions of (p_1, p_2) and (p_3, p_4), we know that p_1 sees p_2 and p_3 sees p_4, and these are the farthest vertices p_1 and p_3 see before n.

Case 1: One of the contradictory lines is the extension of $((n-2), (n-1))$.

Because the region for n is always formed by the line formed by n-1 and its preceding vertex, we can take the extension of $((n-2), (n-1))$ as one of the lines.

Since the region for n is empty, (p_1, p_2) must cross $((n-2), (n-1))$ either before $(n-2)$ or cross the edge between $(n-2)$ and $(n-1)$. However, the first case is not possible, because then (p_3, p_4) has to be entirely to the left of (p_1, p_2), which contradicts the fact the farthest vertex p_1 sees is p_2. In the same way, the second case is not possible, since then p_1 could see $(n-1)$, which leads to the same contradiction.

So, p_1, p_2, $(n-1)$, $(n-2)$ are not distinct points. This means both contradictory lines come from $(n-2)$ or $(n-1)$.

Case 1.1: Let $b = (p_1, (n-2))$ and $a = ((n-2), (n-1))$.

Case 1.1.1: $slope(b) > slope(a)$, so the region is on the left of b and on the right of a. However, it is not possible, because such construction is not inversely complete: $(n-2)$ doesn't see n, p_1 doesn't see $(n-1)$, yet p_1 sees n, which contradicts that the graph is inversely complete.

Case 1.1.2: $slope(b) < slope(a)$. Can't happen, because then the farthest p_1 would see is $(n-1)$.

Case 1.2: Both contradictory lines come from $(n-1)$. Let p_1 be some point before n that sees $(n-1)$, $p_1 \neq (n-2)$.

Case 1.2.1: $slope((n-2), (n-1)) < slope(p_1, (n-1))$. Not possible, since then p_1 wouldn't see $(n-1)$.

Case 1.2.2: $slope((n-2), (n-1)) > slope(p_1, (n-1))$. Let b be the extended edge $((n-2), (n-1))$ and a be the extension of $(p_1, (n-1))$. The region is on the left of b and on the right of a. Then p_1 must see $(n-1)$, $(n-2)$ must see n, yet p_1 must not see n, which is a contradiction to inverse completeness (by which p_1 must see n as well, since the two above lines cross).

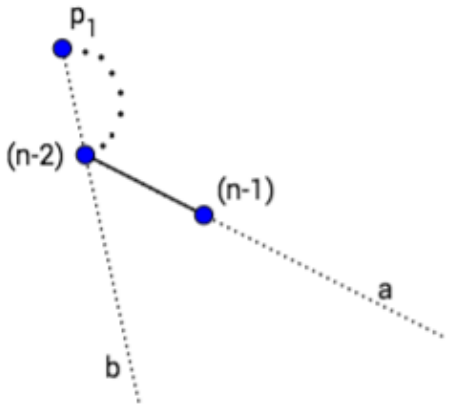

Fig. 3: Case 1.1.1

Case 2: None of the contradictory lines are the extension of $((n-2), (n-1))$.

Let (p_1, p_2) and (p_3, p_4) form two lines forming an empty visibility region. (WLOG, assume that $p_1 < p_2 < p_3 < p_4$, since the proof for the other cases is identical).

Case 2.1: $slope(p_1, p_2) > slope(p_3, p_4)$.

Since the region is assumed to be empty, it must be the case that it is defined on the left of (p_1, p_2) and on the right of (p_3, p_4). The vertex $(n-1)$ must be on the right of both lines, since neither p_1 nor p_3 can see it.

If we attempt to produce a balanced tableau from this construction, it is clear that $T_{p_1p_2} > T_{p_3p_4}$. The farthest vertex p_1 can see before n is p_2, so all the entries in the column containing p_1 between p_1 and n (exclusive) are zeros in $S(T_n)$, and hence are all less than $T_{p_1p_2}$. However, since p_1 sees n, $T_{p_1n} > T_{p_1p_2}$.

Moreover, $T_{p_3n} < T_{p_3p_4} < T_{p_1p_2} < T_{p_1n}$. Yet, T_{p_3n} is a mate of $T_{p_1p_3}$ with respect to T_{p_1n} and and $T_{p_1p_3} \leq T_{p_1p_2}$. However, $T_{p_1n} > T_{p_1p_2}$. Contradiction. This case is not possible if we preserve the ranks of slopes of farthest seen vertices at every stage.

Case 2.2: $slope(p_1, p_2) < slope(p_3, p_4)$.

By symmetry. (Consider the vertex 1 instead of n).

Inductive Hypothesis: Suppose there is a region well-defined (non-empty) constructed by $(k-1)$ lines.

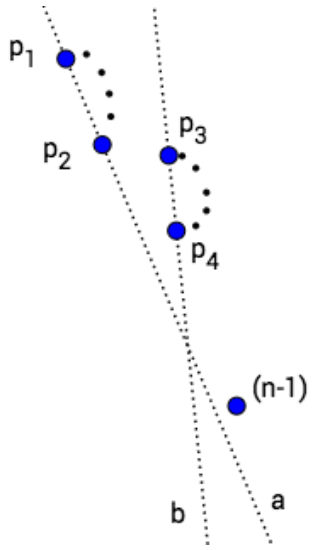

Fig. 4: Case 2.1

Clearly, any non-empty visibility region is a convex set, since it is formed by straight lines.

Add another line defining the region. If the region is still non-empty, we're done, since we can place n in it. Suppose the k^{th} line makes the region for n empty.

Case 1: The region proposed by the line k is on the right of k.

Then the region formed by $\{1, 2, ...k-1\}$ must be on the left of k, since it is assumed to be empty. If all the lines in $\{1, 2, ...k-1\}$ require the region for n to be on the right of them, then the intersection of this region with the region proposed by k is not empty. So $\exists$ a line $m \in \{1, 2, ...k-1\}$ such that the region for n is defined on the left of it. But then m and k are the 2 contradictory lines described in the base case. Contradicts the assumption that the region is empty.

Case 2: The region proposed by the line k is on the left of k. Empty regions are not possible by symmetry with the first case.

□

4.3 *Alternative Proof of the Impossibility of Empty Regions for the General Case*

Proof. As described in the algorithm in [1], if we have a clique balanced tableau, by changing mates in certain order, we can reverse some entries in its skeleton from 1 to 0. Clearly, when we change some 1 to 0 in the skeleton (say for some vertex k), we do not have to preserve its slope rank anymore and the edges between k and its two path neighbors are some consecutive slope ranks we have to preserve (call them x and $(x+2)$).

Clearly, there exists some region for every vertex in a clique matrix. Now we want to show that there exists a region for every reversible entry. When we reverse the first entry in a clique matrix, the vertex it is associated with, k_1, becomes a convex vertex (blocking vertex), k_2. So k_2 still sees everyone k_1 sees. Assuming there was some region for k_1, we want to show there is also a region for k_2. Since the vertex k_1 could see everyone in the clique matrix, its region was defined to the left of all the lines formed by preceding vertices. Since reversing k_1 to k_2, makes k_2 lie on the left of k_1, it has the same region as k_2, and clearly the slope $(x+2)$ can be preserved in this region, since $(x+2) > x$. Then we know the rest of convex vertices for which k_2 is a blocking vertex (before the next blocking vertex) have non-empty visibility regions, by symmetry with the concave-concave special case.

For the remaining reversible entries (not the first one), some reversible v_2 still sees everything v_1 sees, but it might now see more. So if there is a line passing between v_1 and v_2 that defines v_1 on the right, then v_2 would see the vertex this line emanates from that v_1 doesn't see. However, since slope ranks are consecutively preserved ranks, such line doesn't exist and again v_1 and v_2 have the same region. So every blocking vertex has a non-empty region and the rest of the vertices do as well, by symmetry with the concave-concave example.

□

5 Preserving Slope Ranks

Theorem 5.1. *It is always possible to preserve slope ranks of lines connecting a vertex with its farthest seen vertex as we iteratively place a new vertex in its visibility region in the algorithm described above.*

Proof. By induction on the number of vertices n.

Base Case: The base case is trivial. Suppose we have two adjacent vertices 1 and 2 and want to place 3. Clearly, 3 sees 2. If 3 does not see 1, we do not have to preserve the slope, and we're done. Suppose 3 sees 1. Then the slope rank of the edge $(1, 3)$ is greater than that of $(1, 2)$. Also, the region for 3 is defined to the left of the line defined by 1 and 2. The slope of a line connecting 1 and any point on the left of the line satisfies the above condition. One can also check that the rank of the slope of the edge $(2, 3)$ is automatically preserved.

Inductive Hypothesis: Suppose we built a graph on $(n-1)$ vertices preserving needed slope ranks and we want to place the n^{th} vertex.

We want to show that it is possible to place n in its visibility region such that the slope ranks of the edges between n and the vertices it sees are preserved. In the algorithm, as we move from vertex to vertex, we record slopes of edges between farthest seen vertices, according to their ranks. Hence, we can think of preserving slope ranks for a certain vertex as restricting its visibility region using slopes we already recorded, such that the edges between the new vertex and the vertices it sees fall "in the right gaps" between some slopes we already know. So proving that we are always able to preserve slope ranks of edges between n and its visible vertices is equivalent to showing that the visibility region restricted as described above is never empty.

By the lemma above, we know there exists some non-empty region for n. If there exists a point p in that region such that slope ranks can be preserved for the edges between this point and all the vertices n sees, then let $p = n$ and we're done. Suppose that this is not the case, and there exists a point k such that the slope rank of (k, n) cannot be preserved for any placement of n in its region. Then there exist two rays with ranks R and r emanating from k (representing recorded slopes), such that $r < rank(k, n) < R$, and the intersection of these rays and the region of n is empty.

Case 1: R and r are on the left of the visibility region for n. Since the area between the rays and the visibility region are assumed to be disjoint, $\exists$ at least one line m defining the region for n such that $m < r$.

Suppose r emanates from p_1.

Case 1.1: r defines the region for n. Since r is on the left of the region for n, the region must be defined on the right of r, otherwise it would be empty. So p_1 doesn't see n, and we do not have to preserve the slope rank of this edge. Same argument applies if R defines the region for n.

Case 1.2: None of the rays R, r define the region for n. Yet, R and r were recorded at some point, so they represent 1-entries in $S(T_n)$. Since the region for n is formed by the latest 1-entries before n, R and r are not the latest 1-entries before n (i.e. each of them has at least another 1-entry below it which forms the visibility region for n). By the Local Maxima Rule, this implies the slope ranks of all lines forming the region for n are greater than the slope ranks of both R and r. Yet, this is a contradiction, since $m < r$.

Case 2: R and r are on the right of the visibility region for n.

Case 2.1: R defines the region for n. Then R is defined by some p_1 and p_2. The region has to be defined on the left of the R, otherwise we would get an empty region for n. Then p_1 sees n, and we have to preserve the rank of (p_1, n). But then $(p_1, n) > (p_1, p_2)$, which contradicts that $rank(p_1, n)$ falls in-between R and r.

Case 2.2: R does not define the region for n. Then, like in the previous case, it is one of the 1-entries above the 1-entries defining n. But then (p_1, n) has an upper bound rank less than itself. Contradiction.

□

6 Conclusion

We have presented a deterministic algorithm that recovers a staircase polygon from a slope-ranking balanced tableau and proved that certain conditions required by the algorithm always hold. It may be interesting to find an easier proof of impossibility of empty regions for the general case; for example, by

generalizing the proofs for the special cases we considered. It may also be interesting to investigate the optimality of the algorithm discussed above.

ACKNOWLEDGMENTS

The author would like to acknowledge, with gratitude, the direction and support of her research mentor Professor James Abello. She would also like to thank Patrick Chen, a participant of DIMACS 2016 REU, whose code and report she used in her work.

This work was carried out while the author was participant in the 2017 DIMACS REU program at Rutgers University, supported by NSF grant CCF-1559855.

REFERENCES

[1] J. Abello, O. Egecioglu, K. Kumar, "Visibility Graphs of Staircase Polygons and the Weak Bruhat Order I: From Visibility Graphs to Maximal Chains" , Discrete and Computational Geometry, Vol. 14, No 3, 1995, pp 331-358.