A provably-fast linear-expected-time maxima-finding algorithm

Mordecai Golin

▶ **To cite this version:**

Mordecai Golin. A provably-fast linear-expected-time maxima-finding algorithm. [Research Report] RR-1470, INRIA. 1991. inria-00075092

HAL Id: inria-00075092
https://inria.hal.science/inria-00075092

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 1470

Programme 2
Calcul Symbolique, Programmation
et Génie logiciel

A PROVABLY-FAST LINEAR-EXPECTED-TIME MAXIMA-FINDING ALGORITHM

Mordecai J. GOLIN

Juin 1991

A Provably-Fast Linear-Expected-Time Maxima-Finding Algorithm

Mordecai J. Golin*
INRIA-Rocquencourt

May 17, 1991

Abstract: Bentley Carkson and Levine recently presented a new algorithm, the Move-to-Front algorithm, for finding the maximal points in a set. The conjectured that, for n points drawn from any distribution with the component independence property, their algorithm runs in $n+o(n)$ expected time. We prove their conecture.

Un algorithm rapide de recherche de maxima en temps linéaire

Résumé : Récemment Bentley, Clarkson et Levine ont presenté un algorithme, l'algorithme *Move-to-Front*, pour trouver les points maximaux d'un ensemble de n points. Nous établissons que, conformément à leur conjecture, l'algorithme a une complexité moyenne de $n+o(n)$ sur n points tirés avec une distribution ayant la propriété d'indépendance de composantes.

*This work was supported in part by NSF Grant DCR-8605962 and Office of Naval Research Grant N00014-87-K-0460. It was later supported by a Chateaubriand fellowship from the French Ministère des Affaires Étrangères and by the European Community, Esprit II Basic Research Action Program Number 3075 (ALCOM). Author's current address: Algorithms' Project, Inria Rocquencourt, 78153, Le Chesnay Cedex, France.

A Provably-Fast Linear-Expected-Time Maxima-Finding Algorithm

*Mordecai J. Golin**

ABSTRACT

In a recent paper Bentley, Clarkson, and Levine presented some fast (low multiplicative constants) linear expected-time algorithms for finding the maxima of N points chosen I.I.D. from a Component Independent distribution. They also presented another algorithm, the Move-To-Front algorithm, which empirical evidence suggests runs faster than the other algorithms. They conjectured that the Move-To-Front algorithm runs in $N+o(N)$ expected time. In this paper we prove their conjecture for N points chosen I.I.D. from any two-dimensional Component Independent distribution. The proof mixes probabilistic and amortized techniques.

1. Introduction

In a recent paper Bentley, Clarkson, and Levine [1990] presented a collection of new algorithms for finding the maxima of point sets in fast expected time. One of their algorithms, a Move-to-Front heuristic, ran faster than their other algorithms. Based on empirical evidence they conjectured that this MTF algorithm runs in $N+o(N)$ expected time on inputs of N points chosen Independently, Identically, Distributed (I.I.D.) from any distribution that possesses the Component Independence (CI) property. Thus, in a probabilistic sense, this algorithm is asymptotically optimal. furthermore the algorithm is simple to code and seemingly robust in practice.

In this paper we will prove the validity of their conjecture when the input points are chosen from any *two-dimensional* Component Independent distribution. We shall in fact prove something stronger, that with very high (super-polynomial) probability, the MTF algorithm will run in $N+o(N)$ time.

Section 2 of this paper introduces the MTF algorithm and the empirical evidence that led Bentley, Clarkson and Levine to their conjecture. Section 3 presents our analysis. Section 4 concludes the paper by addressing some remaining open questions.

2. The Algorithm

A two dimensional point $p=(p.x,p.y)$ is said to *dominate* another two dimensional point $q=(q.x,q.y)$ if $p.x \geq q.x$ and $p.y \geq q.y$. Similarly when $p=(p_1,p_2,\cdots,p_k)$ and $q=(q_1,q_2,\cdots,q_k)$ are k-dimensional points we say that p dominates q if $p_i \geq q_i$ for all $i \leq k$.

A point p_i is said to be *maximal* in a set of points $p_1,p_2,\cdots,p_n$ if no point p_j, $p_j \neq p_i$, dominates p_i. The maximal points are known collectively as the *maxima*.

Maxima occur quite naturally in many applications. Examples abound in statistics, economics, and graphics. For further details see Preparata and Shamos' book [1985]. The algorithmic question is how to find the maxima of a point set efficiently.

First we need some background. It is possible to show that finding the maxima of a set of N points in two dimensions requires $\Omega(N\log N)$ time in the comparison tree model of computation. For details see

---

* Princeton University. This work was supported in part by NSF Grant DCR-8605962 and ONR Research Grant N00014-87-K0460. Author's current address: Projet Algo, INRIA Rocquencourt, 78153, Le Chesnay Cedex, France.

Preparata and Shamos [1985] who also discuss an algorithm for finding maxima that matches this lower bound; it requires only $O(N\log N)$ comparisons. Kung, Luccio, and Preparata [1975] present an algorithm for finding maxima in K-dimensions. It runs in $O(N\log^{K-2}N + N\log N)$ time and is extremely complicated to implement.

Deterministic running times are not our major concern here. In this paper will be interested in an algorithm with a fast expected running time. In their recent paper Bentley, Clarkson, and Levine [1990] present an algorithm that runs in $O(N)$ expected time on a set of N points chosen I.I.D. from a unit square whose sides are parallel to the cartesian axes. They then extend their algorithm so that it runs in $O(N)$ expected time whenever the points are chosen I.I.D. from an arbitrary distribution that has the Component Independence (CI) property.

A two-dimensional distribution is said to have the Component Independence property if, for a point (x,y) chosen from the distribution (i) the x and y components are independent random variables and (ii) the x and y components are chosen from continuous distributions. For example, the x component might be chosen from an $N(0,1)$ normal distribution and the y coordinate from a Cauchy distribution.

After presenting this second algorithm Bentley, Clarkson and Levine comment that it was "designed to be efficient for CI inputs and easy to analyze for that case but not necessarily robust for point sets from other distributions." They then continue "We will now study an algorithm that is easy to implement, very efficient for CI distributions, and somewhat robust for other distributions." The algorithm referred to in the second quote is the Move-To-Front (MTF) algorithm. We present their pseudo-code implementing the algorithm. It is simple, robust, and easily coded.

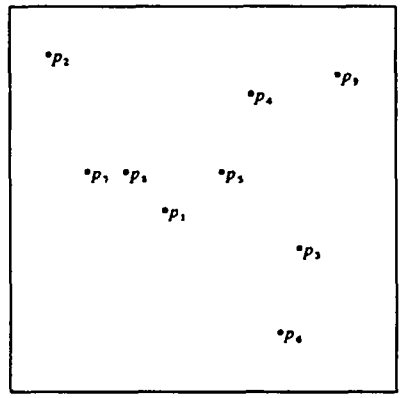
```
TopMax := 1
Max[TopMax] := 1
for I := 2 to N do
  J := 1
  while J <= TopMax do
     if point Max[J] dominates point I then
       move Max[J] to front of Max[1..J]
       J := TopMax + 2
     else if point I dominates point Max[J] then
       shift Max[J+1..TopMax] to Max[J..TopMax-1]
       TopMax := TopMax - 1
     else if point I equals point Max[J] then
       J := TopMax + 2
     else // points I and Max[J] are incomparable
       J := J+1
  if J = TopMax + 1 then
     TopMax := TopMax + 1
     Max[TopMax] := I
```

The Move-To-Front maxima algorithm.

The algorithm scans the points $p_1, p_2, \cdots, p_N$ in sequential order maintaining an ordered list T that contains the current maximal elements among the currently scanned points. In the pseudocode, T is maintained in the array $MAX[\,]$. The algorithm updates the list by comparing p the current point being scanned, to the maxima in the list in left-to-right order.

If p dominates some point q already in the list T then q is no longer maximal and is discarded. If p is not dominated by any point in the list then p is currently maximal and is placed at the end of the list. Otherwise p is dominated by some maximal element, q, in the list. This point q is moved to the front of the list.

We present a fully worked example: The points are as shown in the diagram.The table illustrates how the contents of T change after examining point p_i.



i	New T	Action
1	$[p_1]$	Start
2	$[p_1,p_2]$	Insertion of p_2
3	$[p_1,p_2,p_3]$	Insertion of p_3
4	$[p_2,p_3,p_4]$	Insertion of p_4; p_1 discarded
5	$[p_4,p_2,p_3]$	MTF rule applied
6	$[p_3,p_4,p_2]$	MTF rule applied
7	$[p_4,p_3,p_2]$	MTF rule applied
8	$[p_4,p_3,p_2]$	MTF rule applied
9	$[p_2,p_9]$	Insertion of p_9; p_4, p_3 discarded

How fast is the MTF algorithm? Obviously the algorithm runs in $O(N^2)$ time since T can contain at most $i-1$ points when p_i is examined. It is not hard to construct a degenerate two-dimensional example where $i-1$ comparisons are all made, e.g. when the points $(p.x,p.y)$ are all on the line $p.x=-p.y$.

We are more concerned with discovering the *expected* running time of the algorithm. As a first step in the average case analysis of the MTF algorithm's running time Bentley, Clarkson and Levine observed that the total number of point comparisons performed by it is bounded from above by $\sum_{i<n} M_i$ where M_i is the number of maximal elements in the set $p_1,p_2,\ldots,p_i$:

$$M_i=|\{p_i : p_i \text{ is maximal in} p_1,p_2,\cdots,p_i\}|$$

It is known $E(M_i)=\Theta(\log^{d-1} i)$ for N points chosen I.I.D. from any K-dimensional CI distribution Bentley, Kung, Schkolnick, and C.D. Thompson [1978], Buchta [1989]. Therefore if the points are chosen I.I.D. from such a distribution the expected running time of the algorithm will be upperbounded by

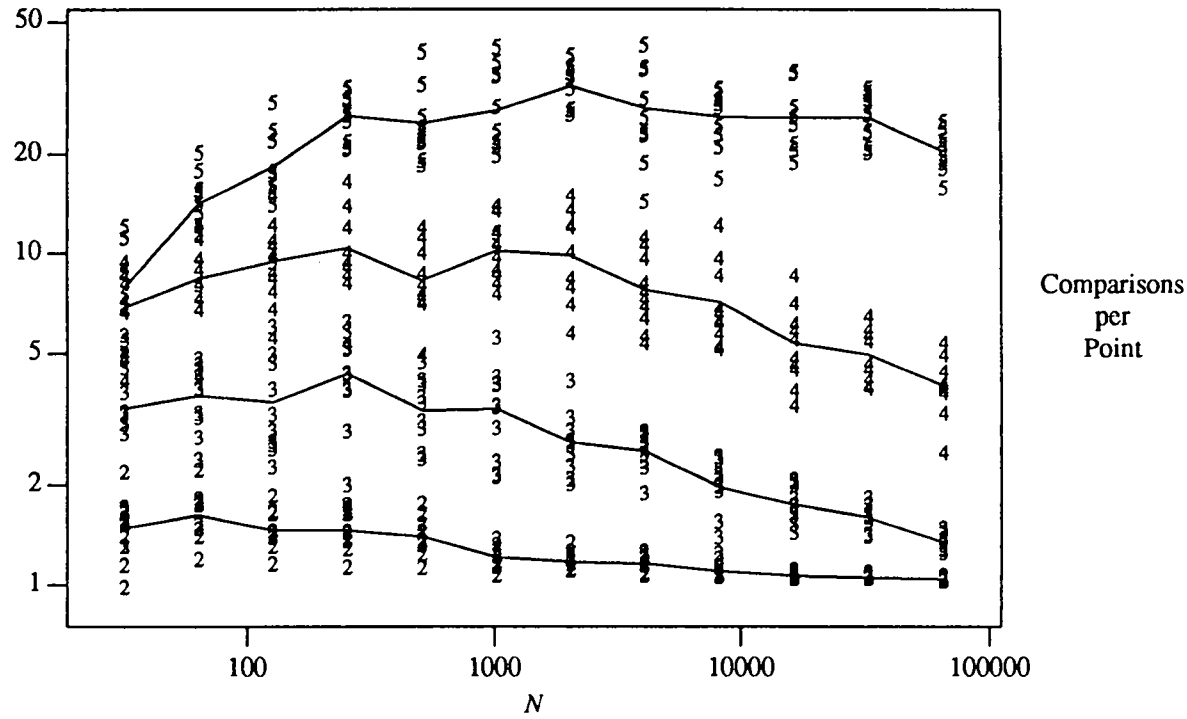
$$E(\sum M_i)=\sum \Theta(\log^{d-1} i)=\Theta(N\log^{d-1} N).$$

They hoped that the algorithms's expected running time would actually be much faster. This hope is plausible because it is possible that Move-To-Front rule would keep "powerful" dominators (points that dominate most other points) at the front of the list. Thus most of the p_i would be dominated by a point near the front of the list. In the worked example p_4 was such a powerful dominator.

Bentley, Clarkson, and Levine ran empirical tests to study the actual behavior of the algorithm on random inputs. We reproduce here two of their graphs containing the results of their tests. The graphs are the result of generating 10 point sets of size N in dimensions 2, 3, 4, and 5. where N runs over all powers of 2 from 32 to 65536. The points were chosen "randomly" from a K dimensional hypercube. Both

dimensions are plotted using logarithmic scales.

The first graph plots the number of point comparisons used divided by N, i.e. the average number of comparisons per point. The plot symbol is the dimension K and the lines pass through the mean values (taken over the ten sets).



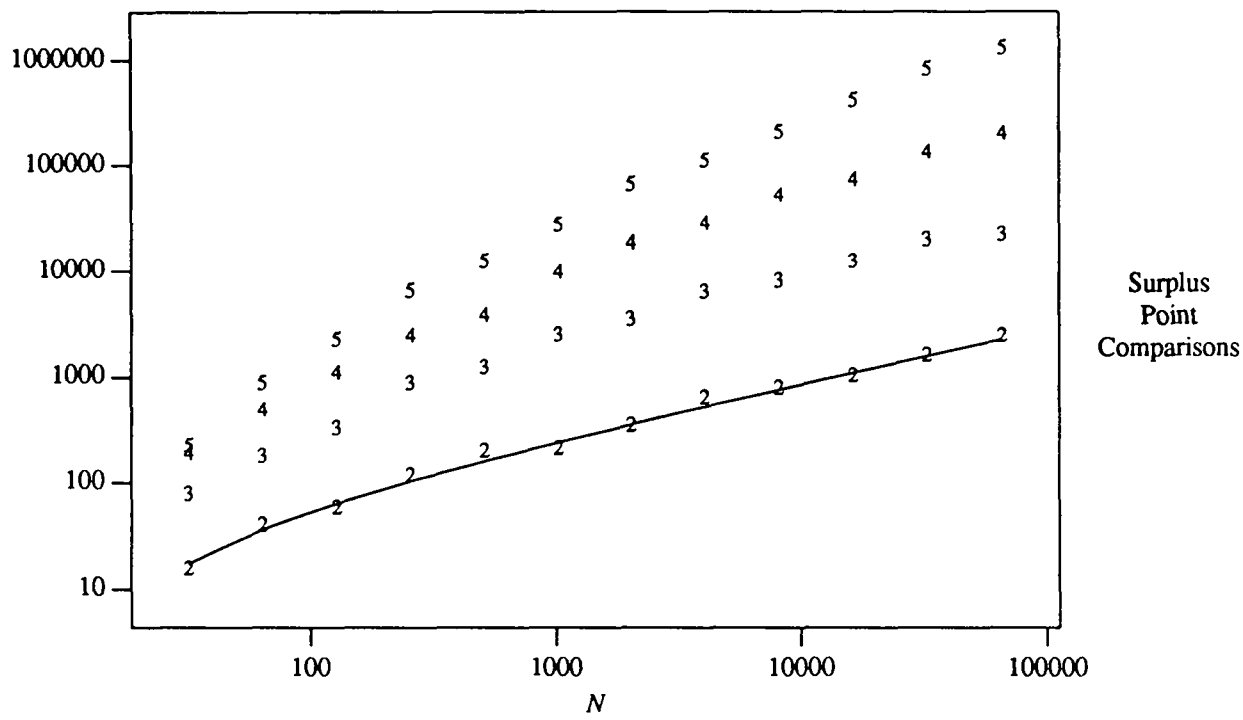
Comparisons per point.

This graph led them to their first conjecture, that the average number of point comparisons performed by the MTF algorithm when examining N points is asymptotic to $C_K N$ where C_K is a constant dependent on K. After further experiments they decided to examine the number of "surplus comparisons." The MTF algorithm must compare each point it examines (except the first) to at least one other point, the point at the front of T) The number of surplus point comparisons is the total number of comparisons minus N; the number of comparisons used over and above the absolute minimum needed. They plotted the number of surplus comparisons in the second graph (on the next page).

A weighted least squares regression showed that for $K=2$ the number of surplus comparisons grew as $7.589N^{.515}-27.74$. Regressions performed on the data for $K=3$, 4 and 5 also had the number of surplus comparisons being sublinear in N. This prompted the following conjecture.

Conjecture [BCL]. The MTF algorithm finds the maxima of N points chosen from a K-dimensional CI distribution in $O(N)$ time using $N+o(N)$ point comparisons for any fixed dimension K.

In the next section we will prove the validity of this conjecture when $K=2$.



Surplus point comparisons.

3. The Analysis

This is the main section of the paper. In it we shall prove the the validity of BCL's conjecture when $K=2$. More specifically we will prove the following:

Theorem 1. Choose N points, $p_1, p_2, \cdots p_N$, from a 2-dimensional CI distribution. Then, with probability $1-N^{-\Omega(\log N)}$, the MTF algorithm finds their maxima using only $N+O(N^{6/7}\log^5 N)$ point comparisons.

The $N^{-\Omega(\log N)}$ term can be thought of as being a superpolynomially small probability since it is smaller than N^{-d} for any constant d. Regardless of its input, the MTF algorithm never performs more than $N(N-1)/2$ point comparisons. Therefore this theorem resolves BCL's conjecture $K=2$. We express this as

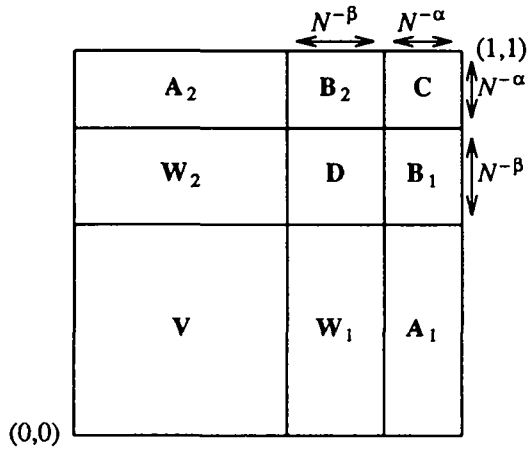
Corollary 1. Choose N points, $p_1, p_2, \cdots p_N$, from a 2-dimensional CI distribution. The MTF algorithm finds their maxima using only $N+O(N^{6/7}\log^5 N)$ expected point comparisons.

An ideal proof of Theorem 1 would be direct; it would examine each of the p_i and show directly that the MTF algorithm requires only $1+O(N^{-1/7}\log^5 N)$ expected point comparisons. More precisely, it would show that for each i, p_i has to be compared to only an expected $1+O(N^{-1/7}\log^5 N)$ points in the array T. Unfortunately, we know of no such direct proof. Instead, we provide an amortized proof of Theorem 1, one that distributes the costs of the algorithm among the number of occurrences of certain types of points. Our proof is divided into three parts. In the first part we define geometric regions and global random variables. In the second part we present two lemmas: Lemma 1 bounds the number of point comparisons as functions of the defined random variables. Lemma 2 provides probabilistic bounds on these same random variables. We show that, taken together, these two lemmas imply Theorem 1. Finally, in the third part we prove the two lemmas.

A caveat: in our proof we assume that the input points are uniformly distributed in the unit square. It is not difficult to modify the proof so that it remains valid when the points are chosen from any two-dimensional CI distribution. We will sketch how to do this at the conclusion of this section.

Proof of Theorem 1. Part (I). Random regions and variables.

Let $p_1, p_2, \cdots p_N$ be the input points listed in the order in which they are examined. We partition the unit square into 9 rectangles dependent upon parameters α, β as shown below.



Formally:

$$W_1 = (1-N^{-\alpha}-N^{-\beta},1-N^{-\alpha}) \times [0,1-N^{-\alpha}-N^{-\beta}), \quad W_2 = [0,1-N^{-\alpha}-N^{-\beta}) \times (1-N^{-\alpha}-N^{-\beta},1-N^{-\alpha}),$$
$$B_1 = (1-N^{-\alpha},1] \times (1-N^{-\alpha}-N^{-\beta},1-N^{-\alpha}), \quad B_2 = (1-N^{-\alpha}-N^{-\beta},1-N^{-\alpha}) \times (1-N^{-\alpha},1],$$
$$A_1 = [1-N^{-\alpha},1] \times [0,1-N^{-\alpha}-N^{-\beta}], \quad A_1 = [0,1-N^{-\alpha}-N^{-\beta}] \times [1-N^{-\alpha},1],$$
$$D = [1-N^{-\alpha}-N^{-\beta},1-N^{-\alpha}] \times [1-N^{-\alpha}-N^{-\beta},1-N^{-\alpha}],$$
$$V = [0,1-N^{-\alpha}-N^{-\beta}) \times [0,1-N^{-\alpha}-N^{-\beta}),$$
$$C = [1-N^{-\alpha},1] \times [1-N^{-\alpha},1].$$

We now introduce random variables which are functions of the input points and the rectangles. Our convention is to represent regions by normal-font capital letters and random variables by bold-faced capital letters. Set

$$\mathbf{A} = |\{p_i : p_i \in A_1 \cup B_1 \cup C \cup B_2 \cup A_2\}|, \quad \mathbf{W} = |\{p_i : p_i \in W_1 \cup D \cup W_2\}|,$$

and

$$\mathbf{F}_C = \min_{1 \le i \le N} \{i : p_i \in C\}, \quad \mathbf{G}_D = \max_{1 \le i \le N} \min_{k \ge 0} \{k : p_{i+k} \in D \text{ or } i+k = n\}.$$

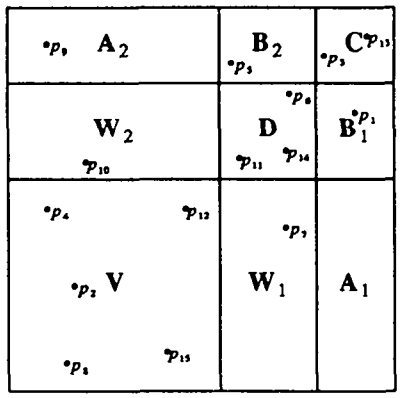
A and **W** are the number of points found in the regions $A_1 \cup B_1 \cup C \cup B_2 \cup A_2$ and $W_1 \cup D \cup W_2$ respectively. Equivalently **A** is the number of points in the strip of width $N^{-\alpha}$ bordering the upper and right sides of the square; as we shall see later this strip will contain all of the maximal points. **W** is the strip of width $N^{-\beta}$ bordering the strip of width $N^{-\alpha}$; as we shall see later most of the comparisons performed by the Move-To-Front rule can be attributed to the points in this strip.

$\mathbf{F}_C$ is the index of the first point in C. $\mathbf{G}_D$ is the maximum gap between points of D. That is, $\mathbf{G}_D$ is the maximum number of points that we must scan to the right of any point p_i before we see the next point in D. This is best understood by example. The four random variables described can best be understood by example.

In the diagram we have $N = 15$, $p_5 \in D$, $p_6 \in D$, $p_{11} \in D$, and $p_{14} \in D$ with none of the other points in D. Therefore $\mathbf{G}_D = 11 - 6 = 5$. In this example $\mathbf{A} = 5$, $\mathbf{W} = 5$ and $\mathbf{F}_C = 3$. We will also need two non-geometric random variables.

$$\mathbf{M}_i = |\{j : j \le i,\ p_j \text{ is maximal in } p_1, \ldots, p_i\}|, \quad i \le N, \quad \text{and} \quad \mathbf{M} = \max_{i \le N} \mathbf{M}_i$$

Example 1:



p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}	p_{13}	p_{14}	p_{15}
B_1	V	C	V	B_2	D	W_1	V	A_2	W_2	D	V	C	D	V

$\mathbf{M}_i$ is the number of maxima in the point set $\{p_1,...,p_i\}$, i.e. the value of *TOPMAX* immediately before p_{i+1} is read. Thus, at most $\mathbf{M}_i$ point comparisons can be performed while comparing p_{i+1} to the current maxima in the course of determining if p_{i+1} is maximal. $\mathbf{M}$ is the maximal value among the $\mathbf{M}_i$; it is thus an upper bound on the number of point comparisons that can be performed while examining any point to see if it is maximal. We will use this last fact quite often. In Example 1 we have $\mathbf{M}=3$ which is achieved by $M_9=M_{10}=M_{11}=M_{12}=3$.

One final piece of notation: recall that the sequence T contains all of the current maxima and is stored in the array *MAX*[ ]. As before, we will represent T as an ordered sequence, e.g. $T=[q_1,q_2,q_3]$. This will denote $T[1]=q_1$, $T[2]=q_2$ and $T[3]=q_3$. We will also use natural generalizations of this notation. For example, $T=[a,b,...]$ will represent a sequence whose first point is a, second point is b, and has an arbitrary number of points in its remaining places.

Part (II). Statements of Lemma 1 and Lemma 2 and proof of Theorem 1.

Lemma 1. The number of point comparisons performed by algorithm M3 when run on a sequence of N points $p_1,p_2,...,p_N$ is at most

$$N + \mathbf{MF}_C + \mathbf{MAG}_D + \mathbf{MW} + \mathbf{MA}.$$

Lemma 2 Let $\alpha > \beta > 0$. Let $p_1,p_2,...,p_N$ be a sequence of N points chosen independently identically distributed from the unit square. Then

$$Pr(\mathbf{A} > 6N^{1-\alpha}) < N^{-\Omega(\log N)},\ Pr(\mathbf{W} > 6N^{1-\beta}) < N^{-\Omega(\log N)},$$

$$Pr(\mathbf{F}_C > N^{2\alpha}\log^2 N) < N^{-\Omega(\log N)},\ Pr(\mathbf{G}_D > N^{2\beta}\log^2 N) < N^{-\Omega(\log N)},$$

$$Pr(\mathbf{M} > 2\log^3 N) < N^{-\Omega(\log N)}.$$

The constants implicit in the *OMEGA*( ) notation are dependent only on α and *beta*.

These two lemmas separate the deterministic part of the analysis from the probabilistic part. Inserting the probabilistic bounds of Lemma 2 into the deterministic one of Lemma 1 yields that algorithm M3 performs

$$N + O(\max(N^{2\alpha}\log^5 N,\ N^{1-\alpha+2\beta}\log^5 N,\ N^{1-\beta}\log^3 N)) \qquad (*)$$

point comparisons with probability $1-N^{-\Omega(\log N)}$. To get the best possible bound we minimize this expression by (ignoring the log factors and) setting

$$2\alpha = 1 - \alpha + 2\beta = 1 - \beta.$$

Solving these three equations simultaneously yields $\alpha = 3/7$, $\beta = 1/7$. Substituting these values back into (*) shows that, with probability $1 - N^{-\Omega(\log N)}$, the MTF algorithm finds the maxima of N points drawn uniformly from the unit square using only $N + O(N^{6/7}\log^5 N)$ point comparisons and proves Theorem 1.

Part (III). Proofs of Lemmas 1 and 2

(A) Proof of Lemma 1:

Plan: To prove this lemma we will partition the input sequence $p_1, p_2, \ldots, p_N$ into a number of contiguous subsequences. We will then show that the number of point comparisons performed while examining each subsequence can be bounded by functions of the random variables defined in Part (I). To be more specific: we will define variables i_k and t_k such that

$$i_0 < t_1 \le i_1 < t_2 \le i_3 < \cdots < t_k \le i_k$$

We will show that

The total number of point comparisons needed to examine $p_1, p_2, \cdots, p_{i_0}$ is at most $\mathbf{MF}_C$.

The total number of point comparisons needed to examine $p_{i_{j-1}+1}, p_{i_{j-1}+2}, \cdots, p_{t_j}$ summed over all j is at most $\mathbf{MAG}_D$.

The total number of point comparisons needed to examine $p_{t_j+1}, p_{t_j+2}, \cdots, p_{i_j}$ summed over all j is at most $N + \mathbf{M}W + \mathbf{M}A$.

The proof of the lemma will follow from these three facts.

Proof: We define the initial (0'th) subsequence to consist of all points up to and including $p_{\mathbf{F}_C}$: the only point in the 0'th subsequence that is located in region C is its final point, $p_{\mathbf{F}_C}$. The other subsequences are inductively defined as starting immediately after their predecessor ends and containing all points up to and including the next point in $A_1 \cup B_1 \cup C \cup B_2 \cup A_2$. Formally the 0'th subsequence is the sequence

$$p_1, p_2, \cdots, p_{i_0}$$

and the j'th subsequence ($j > 0$) is the sequence

$$p_{i_{j-1}+1}, p_{i_{j-1}+2}, \cdots, p_{i_j}$$

where we have set

$$i_0 = \mathbf{F}_C$$

$$i_1 = \min_{i > i_0}\{i : p_i \in A_1 \cup B_1 \cup C \cup B_2 \cup A_2\}$$

and generally

$$i_j = \min_{i > i_{j-1}}\{i : p_i \in A_1 \cup B_1 \cup C \cup B_2 \cup A_2\}.$$

In these definitions we have assumed the existence of $\mathbf{F}_C$. This is not a problem since this Lemma will be used in conjunction with Lemma 2 which implies the existence of $\mathbf{F}_C$ with very high probability. If p_N is not in $A_1 \cup B_1 \cup C \cup B_2 \cup A_2$ then the last subsequence might have to be defined differently than the others. Let $s = \max\{i : p_i \in A_1 \cup B_1 \cup C \cup B_2 \cup A_2\}$. If $s = N$ then the last subsequence is well defined. Otherwise we set the final subsequence to be $p_{s+1}, \ldots, p_N$.

This introduced notation might seem overly complicated but in the end it will simplify our analysis. To make matters more concrete we refer back to example 1. For the points given there $i_0 = 3$, $i_1 = 5$, $i_2 = 9$, $i_3 = 13$, and $i_4 = 15$. Thus the 0'th subsequence is $p_0, \cdots, psub3$, the first subsequence is $p_4, \cdots, p_5$, the second subsequence is $p_6, \cdots, p_9$, the third subsequence is $p_{10}, \cdots, p_{13}$ and the final subsequence is p_{14}, p_{15}. We now make a simple but extremely useful observation.

Claim 1. At most $\mathbf{MF}_C$ point comparisons are performed while examining, $p_1, p_2, \ldots, p_{i_0}$, the points in the 0'th subsequence.

This claim follows immediately from the fact that $i_0 = \mathbf{F}_C$ and the fact **M** is an upper bound on the number of point comparisons performed while examining any point.

The 0'th subsequence is defined differently than the other subsequences because its purpose is different than the other subsequences'. The purpose of the 0'th subsequence is to ensure that T will always contain some current maximal point in C. More specifically, immediately after examining the 0'th subsequence the point P_{F_c} will be inserted into T. Since points in C can only be dominated by other points in C we find that after examining the 0'th subsequence T will always contain some current maximal point located in C. As a consequence, all of the points in the j'th subsequence ($j > 0$) except possibly for i_j are dominated by some point in T and will force the first such point to the front of T.

Next we bound the number of point comparisons that are performed while examining the points in the j'th stage ($j > 0$),

$$p_{i_{j-1}+1}, p_{i_{j-1}+2}, \ldots, p_{i_j}.$$

With the exception of the last point, p_{i_j}, all of the points in the j'th stage are in $W_1 \cup W_2 \cup V \cup D$ and, therefore, by the reasoning of the last paragraph, are dominated by some maximal point in T. For each i, $i_{j-1} < i < i_j$, the number of point comparisons performed while examining p_i is the index in T of the first maximal point that dominates p_i. We would like to show that all points in the j'th subsequence are dominated by the first point in T. Unfortunately, as we shall soon see (Example 3) this does not have to be true.

As a first step we would like to guarantee that the maximal point at the front of T will dominate all points in V. We know that maximal points in in $B_1 \cup C \cup B_2$ dominate all points in V. It makes sense, then, to force such a maximal point to the front of T. Simple geometric reasoning shows that, immediately after examining a point in region D, the maximal point at the front of T must be in $B_1 \cup C \cup B_2$. Therefore we define

$$t_j = \min_{i > i_{j-1}} \{p_i \in D, \text{ or } i = i_j\}, \quad j > 0:$$

t_j is the index of the first point in the j'th subsequence that is located in region D. (Failing the existence of a point in D we set $t_j = i_j$.) After examining p_{t_j} the maximal point at the front of T will be in $B_1 \cup C \cup B_2$ or the algorithm will be finished.

Example 2. We have already seen that $i_0 = 3$, $i_1 = 5$ $i_2 = 9$, $i_3 = 13$ and $i_4 = 15$ for the points in Example 1. Inspection shows that $t_1 = 4$ (because the first subsequence contains no pint in T, $t_2 = 6, t_3 = 10, t_4 = 14$.

We split the j'th subsequence into two parts at the t_j'th point. The first part contains the points $p_{i_{j-1}+1}, p_{i_{j-1}+2}, \ldots, p_{t_j}$ while the second part contains $p_{t_j+1}, p_{t_j+2}, \ldots, p_{i_j}$. The number of points in the first part, $t_j - i_{j-1}$, is at most $\mathbf{G}_D$ where we have already defined

$$\mathbf{G}_D = \max_{1 \le i \le N} \min_{k \ge 0} \{k : p_{i+k} \in D \text{ or } i + k = n\}.$$

Since no point requires more than **M** point comparisons we have just shown that examining the first part of the j'th subsequence requires at most $\mathbf{MG}_D$ point comparisons. Furthermore, the last point of each subsequence j (except possibly the last one) is $p_{i_j} \in A_1 \cup B_1 \cup C \cup B_2 \cup A_2$. There are at most **A** such points p_j thus proving

Claim 2. The total number of point comparisons performed while examining *all* of the points in the first part of *all* of the subsequences is upper bounded by $\mathbf{MAG}_D$.

It now remains to analyze the number of point comparisons performed while examining all of the points in the second parts of all of the subsequences. For this we will rely on a technical lemma:

Lemma 3 Suppose that during the execution of the MTF algorithm the following conditions are simultaneously fulfilled: (i) There is some maximal point $c \in C$ in T. (ii) The sequence T has the form $T=[b,...]$, where $b \in B_1 \cup C \cup B_2$. (iii) The next m points to be examined, $q_1,...,q_m$, are all located in $V \cup W_1 \cup D \cup W_2$. Then the total number of point comparisons performed by the MTF algorithm while examining $q_1,...,q_m$ is upper bounded by $m + \mathbf{M}\overline{\mathbf{W}}$ where

$$\overline{\mathbf{W}} = |\{i : q_i \in W_1 \cup D \cup W_2,\ 1 \le i \le m\}|.$$

Given this new lemma we can easily bound the total number of point comparisons performed while examining the second part of the j'th subsequence, $j > 0$. This is because the second part of the j'th subsequence (except for p_{i_j}) satisfies the conditions of the lemma. Condition (i) is satisfied because after examining the 0'th subsequence some maximal point $c \in C$ is in T. Condition (ii) is satisfied because immediately after examining t_j a maximal point in $b \in B_1 \cup V \cup B_2$ is forced to the front of T. Condition (iii) is satisfied because all of the points $p_{t_j+1}, p_{t_j+2},..., p_{i_j-1}$ are in $V \cup W_1 \cup D \cup W_2$.

We can therefore apply Lemma 3 to to the second part of the j'th subsequence (except for p_{i_j}): $p_{t_j+1}, p_{t_j+2},..., p_{i_j-1}$. Set $m_j = i_j - t_j - 1$, the size of this sequence and

$$\overline{\mathbf{W}_j} = |\{i : p_i \in W_1 \cup D \cup W_2,\ t_j < i < i_j\}|.$$

Application of Lemma 3 shows that the total number of point comparisons performed while examining the points in the second part of the j'th subsequence is upper bounded by $m_j + \mathbf{M}\overline{\mathbf{W}_j} + \mathbf{M}$ (the final $\mathbf{M}$ is the number of comparisons needed to examine p_{t_j}). The subsequences were originally defined as partitioning the entire point set so $\sum_j m_j \le N$ and $\sum_j \overline{\mathbf{W}_j} \le \mathbf{W}$. We also know that there are at most $\mathbf{A}$ subsequences so $\sum_j \mathbf{M} \le \mathbf{MA}$. We have just proven the following

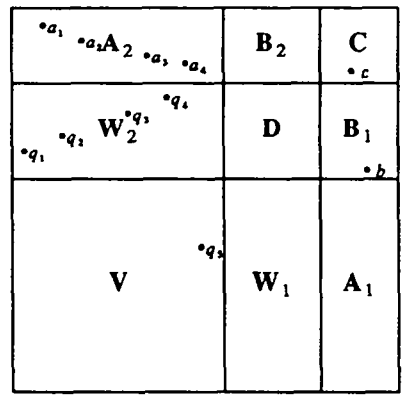
Claim 3. The total number of point comparisons performed while examining *all* of the points in the second part of *all* of the subsequences is upper bounded by $N + \mathbf{MW} + \mathbf{MA}$.

To review: the original N points were partitioned into subsequences and then each subsequence (but the 0'th) was split into two parts. Claim 1 bounds the number of comparisons used to examine the 0'th subsequence. Claims 2 and 3 bound the number of point comparisons needed to examine the other subsequences. Combining the three claims proves Lemma 1 which states that the MTF algorithm requires at most $N + \mathbf{MF}_C + \mathbf{MAG}_D + \mathbf{MW} + \mathbf{MA}$ point comparisons. To finish our proof of Lemma 1 we must verify the validity of Lemma 3.

Proof of Lemma 3: We must show that the total number of point comparisons performed while examining $q_1,...,q_m$ is upper bounded by $m + \mathbf{M}\overline{\mathbf{W}}$ where $\overline{\mathbf{W}} = |\{i : q_i \in W_1 \cup D \cup W_2,\ 1 \le i \le m\}|$. If we could show that all of the $q_i \in V$ are always dominated by the first point in T then we would be finished (since $q_i \notin V$ are counted by $\overline{\mathbf{W}}$). Unfortunately this isn't the case. It is possible to construct situations where points in V require an arbitrarily large number of point comparisons to find a maxima that dominates them. The points are as portrayed in the diagram. The MTF algorithm is ready to examine q_1, q_2, q_3, q_4, q_5. Because of points that it has previously examined the algorithm starts with $T=[b, a_1, a_2, a_3, a_4, c]$. After examining q_1, q_2, q_3, q_4 we find that $T=[a_4, a_3, a_2, a_1, b, c]$. Examining q_5 will require five point comparisons to find c and move it to the front of T.. It is not hard to see how to extend this example so that a point in V will require an arbitrarily large number of point comparisons to find a point that dominates it.

In the notation of Lemma 3 we have $\overline{\mathbf{W}}=4$, $m=5$ and $M=6$. Therefore Lemma 3 tells us that the algorithm will require at most $5+6*4=29$ point comparisons to examine q_1, q_2, q_3, q_4, q_5. In reality, only 19 comparisons are performed.

Example 3.



We should stress two properties of this example. The first is that the a_i-s are moved to the front of T sorted by decreasing x coordinate. The second is that the number of surplus comparisons performed while examining q_5 is not arbitrary. Rather, it is the number of points in $W_1 \cup D \cup W_2$ that were examined since the last time point c was at the front of T.

We will show that these two properties are not particular to this specific example. This fact will enable us to take all of the surplus comparisons performed while scanning $q_1, q_2, \ldots, q_m$ and *amortize* them over the $\overline{W}$ occurrences of points in $W_1 \cup D \cup W_2$. By amortization we mean distributing the total cost of a long sequence of operations over the number of appearances of certain specific occurrences (see Tarjan [1985] for more on the theory of amortization). We now proceed with the proof of Lemma 3. Our proof will be by induction on the length of the sequence $q_1, \ldots, q_m$.

If $m = 0$ then the lemma is obviously true. Assume that the lemma is true for all sequences of size less than $m > 0$. Condition (i) states that before examining the q_i we have $T = [b, \ldots]$ where b is located in one of three different regions: $b \in C$, $b \in B_1$, or $b \in B_2$. If $b \in C$ then b dominates $q_1, \ldots, q_m$ because b dominates the entire region $V \cup W_1 \cup D \cup W_2$: the lemma is trivially true. Otherwise $b \in B_1$ or $b \in B_2$. Without loss of generality we will assume that $b \in B_1$ (the other case is symmetric). We need one more definition which will allow us to describe the possible states that the sequence T can be in.

Definition. The sequence T is said to be in *state* i, $i = 0, 1, 2, \ldots$ if it has the form

$$T = [a_i, a_{i-1}, \cdots, a_1, b, \cdots]$$

where

$$b \in B_1 \cup C,$$

$$a_j \in A_2 \cup B_2, 1 \le j \le i,$$

and

$$a_i.x > a_{i-1}.x > \cdots > a_1.x.$$

Thus T will be in state i if its first i maxima are in $A_2 \cup B_2$ and are sorted by decreasing x-coordinate and its $i+1$'st maximal point is in $B_1 \cup C$.

Example 4: Refer back to Example 3. Before examining q_1, $T = [b, a_1, a_2, a_3, a_4, c]$: T in state 0. After examining q_1 we have $T = [a_1, b, a_2, a_3, a_4, c]$: T in state 1. After examining q_2 we have $T = [a_2, a_1, b, a_3, a_4, c]$ so T is in state 2. Similarly, as q_3 and q_4 are examined, T enters states 3 and 4. Finally, after examining q_5 we have $T = [b, a_4, a_3, a_2, a_1, c]$ and T has returned to state 0. Notice how at each step the state number either increases by one or returns to 0.

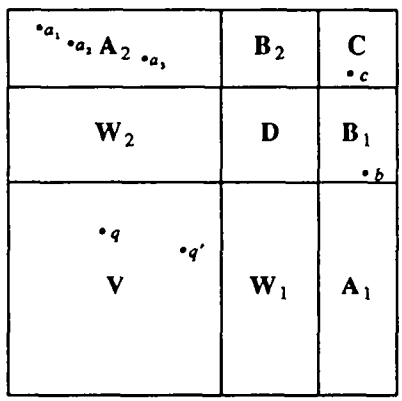
In the new state terminology, condition (ii) of Lemma 3 requires that T be in state 0 before q_1 is examined. We will now show that if T is in state i then examining a new point $q \in V \cup W_1 \cup D \cup W_2$ can only put T into states 0, i or $i+1$. Therefore the defined states include all of the possible configurations that can occur while examining $q_1, q_2, \ldots, q_m$. While proving this we will also count how many point comparisons are performed during the state transitions. Table 1 summarizes the possible outcomes and associated costs. The last column, *point comparisons*, is an upper bound on the number of point comparisons that can be performed.

Outcome	Location of q	Transition	Point Comparisons
(a)	$q \in V \cup W_1$	list remains in state i	1
(b)	"	list returns to state 0	$i+1$
(c)	$q \in D \cup W_2$	list remains in state i	1
(d)	"	list goes to state $i+1$	M
(e)	"	list returns to state 0	M

Table 1: State transitions and their costs

We derive Table 1 by working through what happens when a new point q is examined at a time when $T = [a_i, a_{i-1}, \cdots, a_1, b, \cdots]$ is in state i. First suppose that $q \in V \cup W_1$. There are two possibilities. Either a_i dominates q or it doesn't dominate q. If a_i dominates q then only one point comparison is performed and T remains in state i: this is outcome (a). Otherwise, q is not dominated by a_i. Since $q \in V \cup W_1$ and, for all j, $a_j \in A_2 \cup B_2$ we have, in particular, that $a_i.y > q.y$. Since a_i doesn't dominate q we have $q.x \geq a_i.x$. The definition of state i gives us $q.x > a_i.x > a_{i-1}.x > \cdots > a_1.x$. Thus b is the first point in T that dominates q and $i+1$ point comparisons were performed discovering this fact: outcome (b).

Example 5.

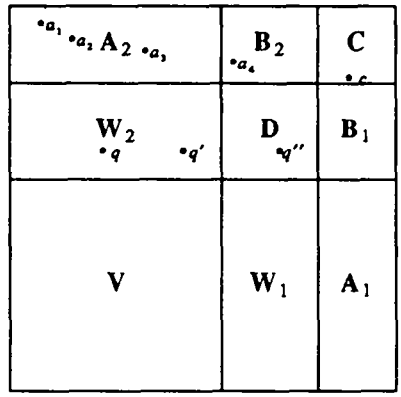


Suppose $T = [a_3, a_2, a_1, b, c]$: T in state 3. The point a_3 dominates q so examining q only requires one point comparison and leaves T unchanged in state 3. This is outcome (a). None of a_1, a_2, a_3 dominate q' so examining q' requires 4 point comparisons and transforms T into $T = [b, a_3, a_2, a_1, c]$: T in state 0: outcome (b).

Now we suppose that $q \in W_2 \cup D$. Again the same two possibilities can occur: a_i dominates q or it doesn't dominate q. If a_i dominates q, then only one point comparison is performed and T remains in state i: outcome (c). Otherwise, as before, we find that $q.x > a_i.x > a_{i-1}.x > \cdots > a_1.x$ and none of the a_j dominate q. We know (condition (i)) that there is some maximal point in T that dominates q and this point will be moved to the front of T. We do not know where in T this point was located so we can only say that it takes at most M point comparisons to find it. If this new maximal point is in $A_2 \cup B_2$ then we label it a_{i+1}. From the definition of domination $a_{i+1}.x > q.x > a_i.x$ so T has entered state $i+1$: outcome (d). If this new maximal point isn't in $A_2 \cup B_2$ then it can't be in A_1 because a point in A_1 dominates no point in

$W_2 \cup D$. It can't be in $W_1 \cup D \cup W_2 \cup V$ because (condition (1)) T contains a point in C. Therefore the new maximal point must be in $B_1 \cup C$: outcome (e).

Example 6.



We assume that $T=[a_3,a_2a_1,a_4,c]$: T is in state 3. The point q is dominated by a_3, thus examining q requires only 1 point comparison and leaves T unchanged: outcome (c). The points q' and q'' are not dominated by a_3 or b but are dominated by a_4 and c respectively. Examining q' transforms T into $T=[a_4,a_3,a_2,a_1,c]$ and T is in state 4: outcome (d). Examining q'' transforms T into $T=[c,a_3,a_2,a_1,a_4]$ and T is in state 0: outcome (e).

We now use these facts about transitions to complete the proof of Lemma 3 The general idea is that the only expensive outcomes are of type (b), (d), and (e). Furthermore, type (b) outcomes are special in that they can be charged to the type (d) outcomes that precede them. Remember our assumptions; that the lemma is true for all sequences of less than m points and that $T=[b,...]$, where $b\in B_1 \cup C$ (T starts in state 0).

First suppose that while examining q_1, T enters state 1 and subsequently, while examining $q_2, \cdots, q_m$ it never returns to state 0. Let i be the state that T is in after examining q_m. Referring to Table 1 we see that all the outcomes that occur must be of type (a), (c), or (d). Type (a) and (c) outcomes cost only one point comparison apiece. Type (d) outcomes can only occur while examining points in $W_2 \cup D$ so there are at most $\overline{\mathbf{W}}$ of these. Furthermore only type (d) outcomes can increase the sate number so $i\le\overline{\mathbf{W}}$. Therefore at most $(m-i)+\mathbf{M}i \le m+(\mathbf{M}-1)\overline{\mathbf{W}}$ point comparisons are performed and the lemma is proven for this case.

Otherwise T does return to state 0 while examining $q_1,...,q_m$. Let m' be the index of the first point that returns T to state 0:

$$m'=\min_{j\ge 1}\{j : \text{The sequence } T \text{ returns to state 0 after examining } q_j\}.$$

By definition, T never returns to state 0 while examining $q_1,...,q_{m'-1}$ so the analysis performed in the previous paragraph tells us that at most $(m'-1)+(\mathbf{M}-1)\overline{\mathbf{W}}'$ point comparisons are performed while examining $q_1,...,q_{m'-1}$ where we define

$$\overline{\mathbf{W}}' = |\{j : q_j\in W_2 \cup D,\ 1\le j<m'\}|.$$

Furthermore, we know that after examining $q_{m'-1}$, T is in some state $i>0$ where $i\le\overline{\mathbf{W}}'$.

We now look at how many point comparisons are performed while examining $q_{m'}$. Since this step returns us to state 0 we know that it must end in an outcome of type (b) or type (e). Let

$$\overline{\mathbf{W}}'' = |\{i : q_i \in W_2 \cup D, 1 \le i \le m'\}|.$$

If $q_{m'} \in V \cup W_1$ then examining $q_{m'}$ leads to outcome outcome (b) with $i+1$ point comparisons being performed. We know that $i \le \overline{\mathbf{W}}' \le \overline{\mathbf{W}}''$. Therefore the number of point comparisons performed while examining $q_1,\ldots,q_{m'}$ is at most

$$(m'-1)+(\mathbf{M}-1)\overline{\mathbf{W}}'+(i+1) \le m'+(\mathbf{M}-1)\overline{\mathbf{W}}'+\overline{\mathbf{W}}' \le m'+\mathbf{M}\overline{\mathbf{W}}''$$

Otherwise, $q_{m'} \in W_2 \cup D$, we see outcome (e), $\overline{\mathbf{W}}'' = \overline{\mathbf{W}}'+1$, and the number of point comparisons performed while examining $q_1,\ldots,q_{m'}$ is at most

$$(m'-1)+(\mathbf{M}-1)\overline{\mathbf{W}}'+\mathbf{M} \le m'+(\mathbf{M}-1)(\overline{\mathbf{W}}'+1) = m'+(\mathbf{M}-1)\overline{\mathbf{W}}''$$

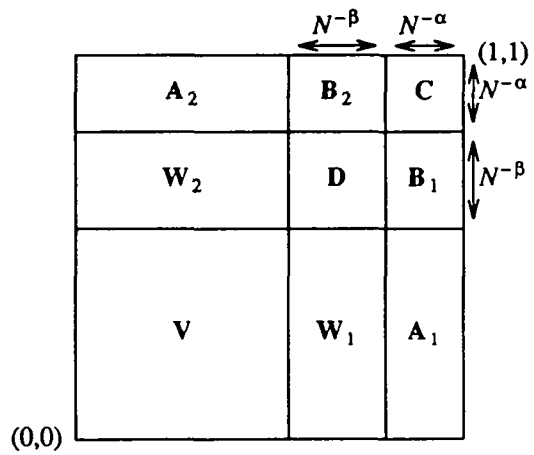
Example 7. Refer back to example three. Let the points be defined as in that example and let T start in state 0, $T=[b,a_1,a_2,a_3,a_4,c]$. In this case we have $m=m'=5$. Immediately before examining q_5 T is in state $i=4$. Since $q_5 \in V$ and examining q_5 returns T to state 0 we know that that examining $qsyb5$ leads to a type b outcome and only requires $i+1=5$ point comparisons.

In both cases we have proven that at most $m'+\mathbf{M}\overline{\mathbf{W}}''$ point comparisons are performed while examining $q_1,\ldots,q_{m'}$. Furthermore, after examining $q_{m'}$, the sequence T is back in state 0. Therefore we can recursively apply the lemma to yield that fewer than $(m-m')+\mathbf{M}(\overline{\mathbf{W}}-\overline{\mathbf{W}}'')$ point comparisons are performed while examining $q_{m'+1},\ldots,q_m$. Combining these two facts proves that at most $m+\mathbf{M}\overline{\mathbf{W}}$ point comparisons are performed while examining $q_1,\ldots,q_m$ and we have proven Lemma 3 for sequences of m points.

(B) Proof of Lemma 2. Let $\alpha > \beta > 0$. Let $p_1,p_2,\ldots,p_N$ be a sequence of N points chosen independently identically distributed from the unit square. We must now prove the probabilistic bounds claimed by Lemma 2:

$$Pr(\mathbf{A} > 6N^{1-\alpha}) < N^{-\Omega(\log N)},\ Pr(\mathbf{W} > 6N^{1-\beta}) < N^{-\Omega(\log N)},$$
$$Pr(\mathbf{F}_C > N^{2\alpha}\log^2 N) < N^{-\Omega(\log N)},\ Pr(\mathbf{G}_D > N^{2\beta}\log^2 N) < N^{-\Omega(\log N)},$$
$$Pr(\mathbf{M} > 2\log^3 N) < N^{-\Omega(\log N)}.$$

These bounds are essentially just variants of classic Chernoff bounds, Hagerup and Rub [1990]. We recall the the random regions introduced at the beginning of this section.



First we will prove

$$Pr(\mathbf{A} > 6N^{1-\alpha}) < N^{-\Omega(\log N)}.$$

Recall that

$$\mathbf{A} = |\{p_i : p_i \in A_1 \cup B_1 \cup C \cup B_2 \cup A_2\}|.$$

Suppose now that q is a point chosen from a uniform distribution over the unit square. Then the probability that $q \in A_1 \cup B_1 \cup C \cup B_2 \cup A_2$ is

$$p = \mathbf{Area}(A_1 \cup B_1 \cup C \cup B_2 \cup A_2) = 2N^{-\alpha} - N^{-2\alpha}.$$

It is not hard to see that $\mathbf{A}$ is a binomially distributed random variable with parameters N and p. Therefore

$$Pr(\mathbf{A} = i+1) = \binom{N}{i+1} p^{i+1}(1-p)^{N-i-1} = \frac{N-i}{i+1}\frac{p}{1-p} Pr(\mathbf{A} = i).$$

When $i > 2Np$ this implies that $Pr(\mathbf{A} = i+1) \le Pr(\mathbf{A} = i)/2$. Telescoping this inequality gives

$$Pr\left[\mathbf{A} = \lfloor 3Np \rfloor\right] \le 2^{1-Np}\, Pr\left[\mathbf{A} = \lfloor 2Np \rfloor\right] < 2^{1-Np}$$

where we use the notation $\lfloor x \rfloor$ to represent the integral part of x. Since $Np = 2N^{1-\alpha} - N^{1-2\alpha} = \Omega(\log^2 N)$ we have

$$Pr\left[\mathbf{A} = \lfloor 3Np \rfloor\right] < N^{-\Omega(\log N)}$$

Telescoping again gives

$$Pr(\mathbf{A} > 3Np) \le Pr\left[\mathbf{A} > \lfloor 3Np \rfloor\right] \sum_{i \ge 0} 2^{-i} < N^{-\Omega(\log N)}.$$

Substituting $p = 2N^{-\alpha} - N^{-2\alpha}$ back into this last equation yields

$$Pr(\mathbf{A} > 6N^{1-\alpha}) < N^{-\Omega(\log N)}.$$

We can prove

$$Pr(\mathbf{W} > 6N^{1-\beta}) < N^{-\Omega(\log N)}.$$

using the same technique. Recall that

$$\mathbf{W} = |\{p_i : p_i \in W_1 \cup D \cup W_2\}|.$$

Replacing p with.

$$p = Pr(q \in W_1 \cup D \cup W_2) = \mathbf{Area}(W_1 \cup D \cup W_2) = 2N^{-\beta} - N^{-2\beta} - 2N^{-\alpha-\beta}$$

the same techniques that we just used to bound $\mathbf{A}$ show that

$$Pr(\mathbf{W} > 6N^{1-\beta}) < N^{-\Omega(\log n)}.$$

We will now prove

$$Pr(\mathbf{F}_C > N^{2\alpha}\log^2 N) < N^{-\Omega(\log N)}, \text{ and } Pr(\mathbf{G}_D > N^{2\beta}\log^2 N) < N^{-\Omega(\log N)},$$

To bound $\mathbf{F}_C = \min_{1 \le i \le N} \{i : p_i \in C\}$ we set

$$p = Pr(q \in C) = \mathbf{Area}(\mathbf{C}) = N^{-2\alpha}.$$

Then

$$Pr(\mathbf{F}_C > j) = Pr(p_i \notin C,\ 1 \le i \le j) = (1-p)^j$$

giving

$$Pr(\mathbf{F}_C > N^{2\alpha}\log^2 N) = (1-p)^{N^{2\alpha}\log^2 N} < N^{-\Omega(\log N)}$$

To bound $\mathbf{G}_D$ we set

$$p = Pr(q \in D) = \mathbf{Area}(D) = N^{-2\beta}.$$

For every $i \le N$ let S_i be the minimum number of points to the right of p_i that we must examine before encountering either a point in D or p_N:

$$S_i = \min_{k>0}\{k : p_{i+k} \in D \text{ or } i+k = N\}.$$

A calculation similar to the one we just performed for $\mathbf{F}_C$ shows that

$$Pr(S_i > N^{2\beta}\log^2 n) < N^{-\Omega(\log N)}.$$

By definition $\mathbf{G}_D = \max_i S_i$ so

$$Pr(\mathbf{G}_D > n^{2\beta}\log^2 N) \le \sum_i Pr(\mathbf{S}_i > n^{2\beta}\log^2 N) < N^{-\Omega(\log n)}.$$

We will now prove the final inequality given by Lemma 2-10,

$$Pr(\mathbf{M} > 2\log^3 N) < N^{-\Omega(\log N)}.$$

Actually, we will prove that, for all $i \le N$,

$$Pr(\mathbf{M}_i > 2\log^3 N) < N^{-\Omega(\log N)}$$

and, since $\mathbf{M} = \max_i \mathbf{M}_i$, the result will follow. To prove this we use the well known fact (see, for example, Golin [1990] for a detailed proof) that $\mathbf{M}_i$, the number of maxima in the set $\{p_1, \cdots, p_i\}$ is known to have the same distribution † as $\mathbf{Z}_1 + \mathbf{Z}_2 + \cdots + \mathbf{Z}_i$ where the $\mathbf{Z}_j$ are independent Bernoulli random variables such that

$$Pr(\mathbf{Z}_j = 1) = 1/j, \quad Pr(\mathbf{Z}_j = 0) = 1 - 1/j.$$

Let $l = \lfloor \log_2 i \rfloor$. We partition the $\mathbf{Z}_j$ into $l+1$ sets of geometrically increasing size by defining new random variables $\mathbf{Y}_k$.

$$\begin{aligned}
\mathbf{Y}_0 &= \mathbf{Z}_1, \\
\mathbf{Y}_1 &= \mathbf{Z}_2 + \mathbf{Z}_3, \\
\mathbf{Y}_2 &= \mathbf{Z}_4 + \mathbf{Z}_5 + \mathbf{Z}_6 + 7_7, \\
&\cdot = \cdot, \\
&\cdot = \cdot, \\
&\cdot = \cdot,
\end{aligned}$$

The general term is

$$\mathbf{Y}_k = \mathbf{Z}_{2^k} + \mathbf{Z}_{2^k+1} + \cdots + \mathbf{Z}_{2^{k+1}-1},$$

while the final term is

$$\mathbf{Y}_l = \mathbf{Z}_{2^l} + \mathbf{Z}_{2^l+1} + \cdots + \mathbf{Z}_i.$$

Since i is usually not of the form $2^{l+1}-1$ we define dummy zero random variables

† For those who are familiar with Stirling numbers this implies that $Pr(\mathbf{M}_i = j) = c(i,j)/i!$ where $c(i,j)$ is a signless Stirling number of the first kind.

$\mathbf{Z}_{i+1}=\mathbf{Z}_{i+2}=\cdots=\mathbf{Z}_{2^{i+1}-1}=0$. This lets us write Y_i in the general form. By definition $\mathbf{M}_i$ has the same distribution as $\sum_{0\le k\le l} \mathbf{Y}_k$ so, to prove that $Pr(\mathbf{M}_i \ge 2\log^3) < N^{-\Omega(\log N)}$, it will suffice to prove, for all k, that

$$Pr(\mathbf{Y}_k \ge \log^2 N) < N^{-\Omega(\log N)}.$$

If $k < 2\log\log N$ then, deterministically, $\mathbf{Y}_k \le 2^k < \log^2 N$. We may therefore assume that $k \ge 2\log\log N$. Using the definition of the Bernoulli random variables $\mathbf{Z}_j$ we know that $Pr(\mathbf{Z}_j = 1) \le 1/2^k$ for all $j \ge 2^k$. Therefore for any integral $c \ge 0$ we can write

$$Pr(\mathbf{Y}_k = c) \le \binom{2^k}{c} \frac{1}{(2^k)^c} < \frac{1}{c!}.$$

If $c \ge \log^2 N$ then Stirling's formula gives

$$Pr(\mathbf{Y}_k = c) < N^{-\Omega(\log N)}.$$

Summing over all $c \ge \log^2 N$ shows that

$$Pr(\mathbf{Y}_k \ge \log^2 N) < N^{-\Omega(\log N)}$$

and thus

$$Pr(\mathbf{M}_i > 2\log^3 N) < N^{-\Omega(\log N)}.$$

End of the proofs of Lemma 2 and Theorem 1.

We have just proven the validity of Theorem 1 when the input points $p_1,\ldots,p_N$ are independently identically distributed chosen from the uniform distribution over the unit square. As promised earlier we will now show how to modify the proof so that it remains valid when the points are independently identically distributed and chosen from *any* two-dimensional CI distribution.

Suppose that $p_1=(x_1,y_1),\ldots,p_N=(x_N,y_N)$ are I.I.D. chosen from some two-dimensional CI distribution. Let F_x and F_y be the continuous distribution functions of the x and y components of the points:

$$Pr(x_i \le s) = F_x(s) \quad \text{and} \quad Pr(y_i \le t) = F_y(t).$$

We now define N, the natural mapping from the support of the CI distribution to the unit square.

$$N((x,y)) = (F_x(x), F_y(y)).$$

This mapping has two very useful properties:

(i) If $p=(x,y)$ is chosen from a CI distribution whose components x,y have distribution functions F_x and F_y then the point $N(p)$ has a uniform distribution over the unit square.

(ii) A point p_i dominates another point p_j if and only if $N(p_i)$ dominates $N(p_j)$.

The validity of Theorem 1 when $p_1,\ldots,p_N$ are chosen from any CI distribution will follow. Property (i) tells us that $N(p_1),\ldots,N(p_N)$ are chosen from the uniform distribution over the unit square. Property (ii) tells us that the MTF algorithm performs the exact same sequence of operations on input $N(p_1),\ldots,N(p_N)$ as on the input $p_1,\ldots,p_N$. We have already proven that the theorem is valid for points which, like $N(p_1),\ldots,N(p_N)$, are chosen uniformly from the unit square. Therefore Theorem 1 holds true for points chosen from any two-dimensional CI distribution.

4. Open Questions

In this paper we proved the following conjecture of Bentley, Clarkson, and Levine [1990]: the MTF algorithm performs only $N+o(N)$ expected point comparisons when run on N points chosen I.I.D. from any two dimensional distribution with the CI property. More specifically, we proved that when the points are chosen from such a distribution then with probability $1-N^{-\Omega(\log N)}$ only $N+O(N^{6/7}\log^5 N)$ point comparisons will be made. The proof depended on an amortized probabilistic technique.

We are left with two open questions. The first is how well the MTF algorithm works when the points are chosen from a non-CI two-dimensional distribution. The second is how well the algorithm works if the points are chosen from a higher dimensional distribution.

Bentley Clarkson and Levine ran their algorithm on random inputs where the points were chosen I.I.D. from a two dimensional ball distribution (points chosen uniformly from a circle). In that case they found empirical evidence to suggest that the MTF algorithm runs in $O(N\log(N)$ expected time on such inputs

Similarly, e ran the MTF algorithm on points chosen from the uniform distribution over the triangle with vertices (0,0), (1,0), and (0,1). Our observations are collected in Table 2 below.

For each value of N we ran the algorithm on 100 sets of N points and averaged the results.

N	Average No. Maxima	(Av. No. Maxima)/$\sqrt{N}$	Point Comparisons	(Point Comp)/N
1024	54.62	1.707	6348.78	6.199980
2048	79.65	1.760	14027.88	6.849551
4096	112.91	1.764	30428.94	7.428940
8192	158.65	1.753	64566.32	7.881631
16384	227.02	1.774	138185.07	8.434147
32768	318.18	1.758	290696.86	8.871364
65536	451.79	1.765	612638.86	9.348127
131072	640.41	1.769	1294415.61	9.875607

Table 2:

First, note that the average number of maxima for N points drawn from this distribution is $\sqrt{\pi N} \approx 1.772\sqrt{N}$.† It is the rightmost column of the table which is most interesting in our context. Here, as in the case of the ball distribution, the MTF algorithm seems to use only $O(N\log N)$ point comparisons. In some ways this triangle is a worst case figure since if N points are chosen I.I.D. uniformly from any convex region the expected number of maxima will be $O(\sqrt{N})$ Devroye [1986], Dwyer [1990]. Perhaps the algorithm always has O(N log N) expected behavior when its input points are chosen IID uniformly from a convex region.

The second, and more interesting, question is whether Theorem 1 can be generalized to show that, with high probability, the number of surplus point comparisons is sublinear no matter what the dimension of the space that underlies the CI distribution. That is, is [BCL]'s conjecture valid for all dimensions? Unfortunately it does not seem possible to extend the techniques used in this paper to solve the more general problem. This is because the techniques made implicit use of the ordering properties of points in two dimensions. More explicitly, Lemma 3 can not be extended to higher dimensions. In Lemma 3 we used the fact that the a_i that were brought to the front of the list by the Move-To-Front rule were brought there sorted by decreasing x-coordinate. This fact does not seem capable of generalization to higher dimensions.

† This result can be derived using standard analytic techniques. We do not derive it here because it is beyond the scope of this paper.

5. Acknowledgements

The author would like to thank Jon Bentley for conversations that led to the analysis presented in this paper.

References

Bentley, J.L., K . L. Clarkson, and D. B. Levine. "Fast Linear Expected-Time Algorithms for Computing Maxima and Convex Hulls," *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Mathematics*, 1990, pp. 179-187.

Bentley, J. L., H. T. Kung, M. Schkolnick and C. D. Thompson [1978]. "On the average number of maxima in a set of vectors and applications", *Journal of the Association for Computing Machinery 25*, 4, October 1978, pp. 536-543.

Buchta, Christian [1989]. "On the Average Number of Maxima in a Set of Vectors," *Information Processing Letters 33*, 10 November 1989, pp. 63-65.

Devroye, L [1986]. *Lecture Notes on Bucket Algorithms*, Berkhauser-Verlag, Boston.

Dwyer, Rex [1990]. "Kindler, Gentler, Average-Case Analysis for Convex Hulls and Random Vectors," *SIGACT News 21'* 2, Spring 1990, pp. 64-71.

Golin, Mordecai J. [1990]. "Probabilistic Analysis of Geometric Algorithms," Thesis, Princeton University.

Hagerup, Torben and Christine Rub [1990] " A Guided Tour of Chernoff Bounds" *Information Processing Letters 33*, 1989/90, pp. 305-308.

Preparata, F. P. and M. I. Shamos [1985]. *Computational Geometry*, Springer-Verlag.

Tarjan, Robert E. [1985]. "Amortized Computational Complexity," *SIAM J. Alg. Discrete Math 6*, 2, April 1985, pp. 28-40.

ISSN 0249-6399