AD-A039 872 YALE UNIV NEW HAVEN CONN DEPT OF COMPUTER SCIENCE F/G 9/2
THE COMPLEXITY OF SEARCHING LINES IN THE PLANE: PRELIMINARY VER--ETC(U)
1975 R J LIPTON, D DOBKIN N00014-75-C-0752
UNCLASSIFIED RR-93 NL

| OF |
AD
A039872

END

DATE
FILMED
6-77

REPORT DOCUMENTATION PAGE

BEFORE COMPLETING FORM

1. REPORT NUMBER: (14) RR-93 ✓
2. GOVT ACCESSION NO.:
3. RECIPIENT'S CATALOG NUMBER:

(62)

4. TITLE (and Subtitle): (6) The complexity of searching lines in the plane: Preliminary version.
5. TYPE OF REPORT & PERIOD COVERED: (9) Research Technical
6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): (10) Richard J. Lipton, David Dobkin
8. CONTRACT OR GRANT NUMBER(s): (15) N00014-75-C-0752, N00014-75-C-0450
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Yale University, Department of Computer Science, 10 Hillhouse Ave, New Haven, CT 06520
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: (11) 1975
11. CONTROLLING OFFICE NAME AND ADDRESS: Office of Naval Research, Information Systems Program, Arlington, Virginia 22217
12. REPORT DATE: (11) JUL 1978
13. NUMBER OF PAGES: (12) 16 p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):
15. SECURITY CLASS. (of this report): Unclassified
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:
16. DISTRIBUTION STATEMENT (of this Report): Distribution of this report is unlimited
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report):
18. SUPPLEMENTARY NOTES:
19. KEY WORDS (Continue on reverse side if necessary and identify by block number): linear tree programs, searching, complexity
20. ABSTRACT (Continue on reverse side if necessary and identify by block number):

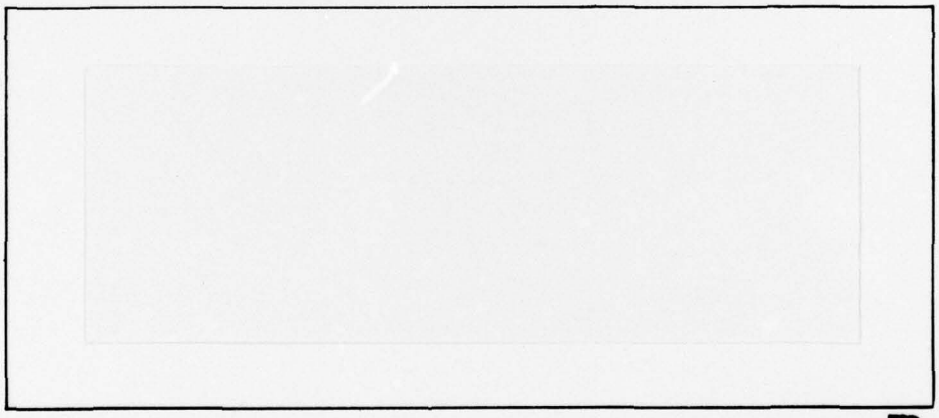
Searching is a fundamental operation of computer science. Yet a number of key mathematical questions about searching in Euclidian spaces remains open. A number of such questions are formulated and answered here for searching lines in the plane. Relationships between the results here and higher dimensional analogs for other problems of interest are given. Among the new results is a mathematical framework in which questions about searching can be stated in a more uniform manner than was possible before. Specific results are also given on the searching complexity of various sets

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

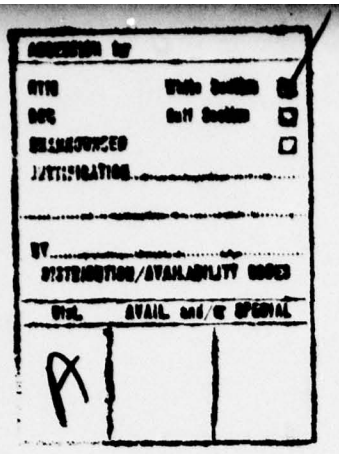
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

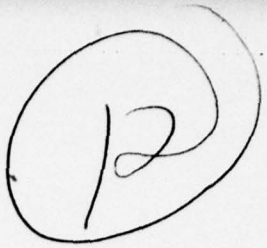
447451

of lines in the plane. In particular, it is shown that there are easy and hard sets of lines to search and establish methods of generating upper and lower bounds on the search complexities of such sets.



YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE



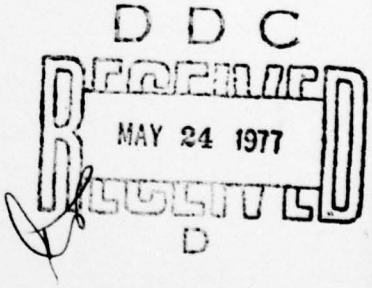


The Complexity of Searching Lines in the Plane: Preliminary Version

David P. Dobkin* and Richard J. Lipton†

Research Report #93

Department of Computer Science
Yale University
New Haven, Connecticut 06520



* Portions of this author's research were supported by ONR Grant N00014-75-C-0450.

† Portions of this author's research were supported by ONR Grant N00014-75-C-0752.

Abstract

Searching is a fundamental operation of computer science. Yet a number of key mathematical questions about searching in Euclidian spaces remains open. A number of such questions are formulated and answered here for searching lines in the plane. Relationships between the results here and higher dimensional analogs for other problems of interest are given. Among the new results is a mathematical framework in which questions about searching can be stated in a more uniform manner than was possible before. Specific results are also given on the searching complexity of various sets of lines in the plane. In particular, we show that there are easy and hard sets of lines to search and establish methods of generating upper and lower bounds on the search complexities of such sets.

I. Introduction

A fundamental operation of computer science is searching. Certainly the majority of actual computation involves the processing and organization of data into sets which are to be sorted in a manner to make repeated searches as simple as possible. Furthermore, Knuth [5] has devoted an entire chapter of his encyclopedic work on computer programming to the study of methods of computer searching. Despite this enormous focus on searching, a number of key mathematical issues regarding searching remain either unexplored or unanswered. Among these issues is the key issue of the searching of a set of geometric objects in Euclidian space. In addition to the existance of such problems as extensions and embellishments to previously studied problems of geometric complexity (see e.g. [2], [9]), this framework appears to be a natural setting for the generation of lower bounds on the knapsack, partition and travelling salesman problems as well as variants of the sorting problem. Furthermore, this methodology has also produced many good upper bounds which can be used to solve practical problems of such diverse areas as information retrieval, numerical analysis, and artificial intelligence. The main goal of this paper will be to lay the beginnings of a unified framework through which all questions of geometric searching can be resolved. To give an idea of the complexity of such a theory we pause to give an example of an elementary result within this theory which appears very anomalous. Consider the problem of determining membership of a point on or among a set of n lines in the plane which are in general position. That is, we are given a set of n lines in the plane with the condition that no three have a point in common and each pair has exactly one point in common (i.e. no two are parallel). We then wish to

ask questions about a new point determining at each query whether it lies to the left of right of one of the given lines. Our procedure halts after enough queries have been made to know whether the given point lies on any of the lines or if not, which lines bound the region in which it lies. A reasonable conjecture, given that any set of n lines of the plane in general position forms exactly $\frac{1}{2}(n^2+n+2)$ regions, is that the searching complexity of any set of n lines in general position is the same. Yet, as we shall see in subsequent sections of this paper there is a set of n lines which can be searched in $O(\log^2 n)$ queries while another set is shown to require n queries, an exponential gap. Such anomalies together with the guiding principle that "intuition about geometric problems is seldom correct" characterize this as a difficult problem. However, recent results [1,2,3,4,9] concerning searching complexities and lower bounds tend to characterize these as fruitful areas of research. Among the results reported in these papers are upper bounds of practical importance on some searching problems as well as lower bounds of $\frac{1}{2} n^2$ and n log n queries on linear search tree programs (i.e. each query is $f(\underline{x})$ R 0 where f is an affine function on the input $\underline{x}$ and R is >, = or < for the knapsack (i.e. Given $x_1,\ldots,x_n$, b does there exist $I \subseteq \{1,\ldots,n\}$ such that $\sum_{i\in I} x_i = b$) and Element Uniqueness (i.e. Given $x_1,\ldots,x_n$, does there exist $i\neq j$ such that $x_i=x_j$) Problems.

In the current paper, we will focus our attention on problems involving searching lines in the plane. Such problems are of interest in themselves as well as a gateway to problems involving hyperplane searches in higher dimensional Euclidian spaces. Our goal will be one of classification of the complexity of searching different sets of lines. Two distinct cases exist, in the first only queries may be made of the original lines and in

the second new lines may be added with queries made with respect to the original or new lines. Thus, if we define $c(A)$ and $\hat{c}(A)$ as the complexity under the first and second measures of searching the set of lines A, then $\hat{c}(A) \triangleq \min_B c(A \cup B)$ where B is any new set of lines. Among the results presented here are

$2 \log |A| \leq \hat{c}(A) \leq 3 \log |A|$ for any set A where $|A|$ is the number of lines in A.

And the existence for each n of sets A_1^n and A_2^n of n lines such that

$$c(A_1^n) \leq 3/4 \log^2 |A_1^n|$$

$$c(A_2^n) = |A_2^n| = n.$$

These results leave us unable to make general statements about the $c(.)$ function as we could about the $\hat{c}(.)$ function. Hence we concentrate our efforts on methods for determining for any set A, the value of $c(A)$. To do so, it is necessary to introduce new ideas to the standard mathematical notions of general position. And it is at this point where our work diverges from the standard mathematical literature on this subject. However, we believe that some of the methods and new ideas introduced here will, in addition to resolving questions regarding searching lines in planes, yield insight into methods of extending the known lower bound of $\frac{1}{2} n^2$ on the complexity of the knapsack problem in n-dimensions, as the issues there are merely higher-dimensional analogs of those introduced here.

The organization of the paper is as follows. In the next section, the exact problem which we are considering is presented in detail. The

concepts briefly spelled out above are concretely defined. Following that, some definitions and results concerning the geometry of intersecting lines in the plane are given. Some of these results belong to the classical mathematical literature on the problem while others were derived within the context of this problem. Results found by applying these results to the problems at hand are also surveyed.

II. Problem Statement

Searching problems in the plane will be our focus. Such a problem consists of a set of lines dividing the plane into regions. Our lines will be in general position, hence no two are parallel and no three have a point in common. Thus the number of regions formed by a set of n such lines will be $\frac{1}{2}(n^2+n+2)$. The searching problem for lines in the plane then consists of determining for a new point in which of these regions it lies. And our goal is to determine the complexity of searching any given set of lines in the plane. The algorithms we allow are linear tree progams which have been widely used before [3,7,10,11]. Such programs consist of three types of statements, branches of the form

S_k: if f(x) R 0 then go to S_m else go to S_n,

and decision statements of the form

S_p: point x belongs to one of the lines

S_q: point x belongs to region R and none of the lines

where f is a linear function on the input point x, R is one of the relations {<,=,>}, and R is a specification of one of the regions formed by the intersecting lines. The complexity of such an algorithm is defined as the longest path from its root to any decision statement.

Within this model, we consider two complexity measures on the searching of lines. In the first, the function f is restricted to represent one of the original lines. Thus, the problem here is to determine to which region a point belongs with only comparison to the original lines. We define the complexity of searching a set of lines, A, under this measure as c(A). One is tempted to believe that c(A) = |A|, the cardinaltiy of A, but the following example shows otherwise.

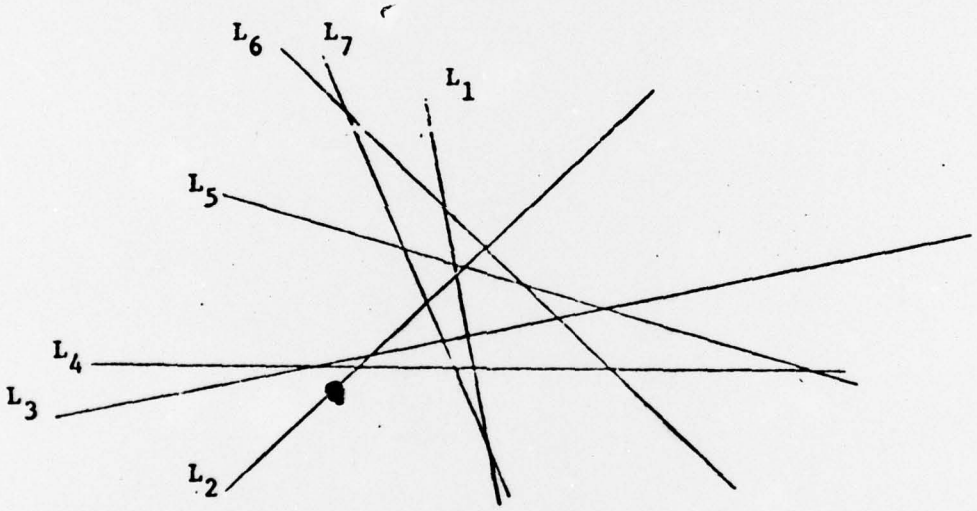


Figure 1: A set of 7 lines to be searched.

We observe that on the left of L_1, the lines L_4, L_5, and L_6 do not intersect and on the right, lines L_2, L_3 and L_7 do not intersect. Hence if x lies to the left of L_1, we can search L_4, L_5, and L_6 by a binary search algorithm and similarly for L_2, L_3 and L_7 if x lies to the right of L_1. Therefore, in at most 6 comparisons we can search these lines. Since all sets of lines

are taken to be in general position, it would be reasonable to assume that c(A) is fixed for fixed |A|. This is untrue, since a set of 7 lines forming a septagon has a searching complexity of 7. We shall see in later sections that c(A) varies greatly with A for fixed |A|.

The second complexity measure we use allows for the introduction of new searching objects. The function f can now be any line in the plane. For this case, we represent the complexity of searching a set A of lines as $\hat{c}(A)$. It is easy to see that $\hat{c}(A) = \min_B c(A \cup B)$ taken over all sets of lines, B. In a previous paper [2], we showed that $\hat{c}(a) \leq 3 \log|A|$ and a simple region counting arguments yields $\hat{c}(A) \geq 2 \log|A|$. However an exact bound on $\hat{c}(A)$ would be of value as this would yield insight into methods of generating better than information theoretic lower bounds on searching. In a related paper, applications of such results to tight bounds on the knapsack problem are studied [4].

Throughout, we shall use r(A) to denote the largest number of sides of any polygon formed by intersetions of the lines in A. Clearly r(A) is a lower bound on c(A).

III. Results

In this section the basic structure of c(A), $\hat{c}(A)$, and r(A) is investigated. In addition to proving a number of simple but basic facts, we also demonstrate that understanding these functions is going to be a non-trivial task. This follows for two diffent but related reasons. First, the classical literature on arrangements of lines in the plane is filled with simple sounding assertions that are open. Indeed much of this literature is still trying to answer questions of the form "how many ... are there?". In

contrast our research requires answers to questions of the form "how many ... are there and where are they with respect to ...?". Second, we are able to prove at least two results that are unexpected. Moreover, these results show that simple and intuitive arguments about even the function c(A) are possibly going to be incorrect. In particular we show that complexity behaves poorly with respect to disjoint union, i.e. there are disjoint sets A and B such that

$$c(A \cup B) << c(A) + c(B)$$

(<< means much smaller. See theorem 5 for details.) This result has a similar flavor to the result of Schnorr [8] on the corresponding result for Bookan circuits.

We first observe the following two easy lower bounds on c(A).

Theorem 1: Let A be a set of lines in the plane. Then $c(A) \geq r(A)$ and $c(A) \geq \log_2|A|$.

Proof: Recall that r(A) is the size of the largest region formed by A. Thus, a simple adversary argument demonstrates the lower bound of r(A). The lower bound of $\log_2|A|$ is the usual information theory argument. □

We now study a simple general method of obtaining upper bounds on c(A), $\hat{c}(A)$, and r(A).

Theorem 2: Let A and B be sets of lines in the plane. Then

(1) $c(A \cup B) \leq c(A) + c(B)$

(2) $r(A \cup B) \leq r(A) + r(B)$.

Proof:

(1) Any search trees for A and B respectively can be combined to form one for A∪B of size at most c(A) + c(B). (This uses the convexity of the regions that A and B form.)

(2) We sketch a proof that $r(A \cup B) \leq r(A) + r(B)$. Let R be the largest region of A∪B; let $r_1,\ldots,r_k$ be the sides of R. Partition $r_1,\ldots,r_k$ into $s_1,\ldots,s_m$ and $t_1,\ldots,t_n$ such that each s_i is part of a line from A and each t_i is part of a line from B. By a convexity argument we can show that there is a region with at least m sides in A (alone) and one with at least n sides in B (alone). Thus, $r(A \cup B) = m+n \leq r(A) + r(B)$. The convexity argument is as follows: Consider the sides $s_1,\ldots,s_m$. Now extend them; they form a region with m sides. The other lines from A can not cut any of $s_1,\ldots,s_m$ by definition; hence, in A we must have a region of at least m sides. □

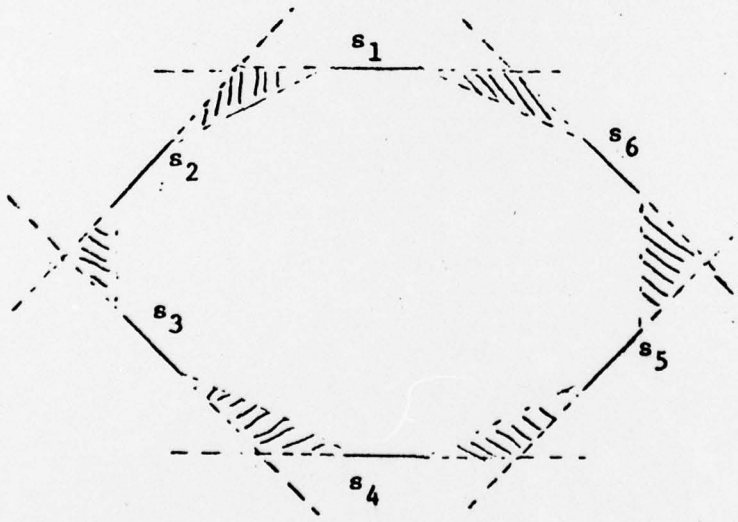


Figure 2: Only the shadowed regions can be cut by other lines of A; hence, a region of ≥m sides exists in A.

If the last theorem were tight one might hope that c(A) would be about $|A|$. This is of course trivially false if A contains parallel lines. Thus a more interesting question is: Does c(A) equal about $|A|$ for A in general position? We know from [2] that for $\hat{c}(A)$ this is false, i.e.

$$\hat{c}(A) \leq 3 \log_2 |A|.$$

We now show that c(A) <u>can</u> also be very small compared to $|A|$. Note before we continue that $c(A) = |A|$ is possible for A in general position since there are such A with $r(A) = |A|$.

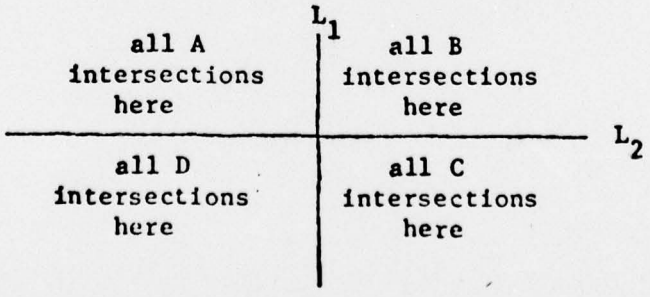
<u>Theorem 3</u>: For any n there is a set of lines $|A| = n$ in general position such that $c(A) = O(\log^2 |A|)$.

<u>Proof</u>: We proceed by induction. Let k be a constant such that for each $i<n$, there is a set, x_i, of i lines with $c(x_i) \leq k \log^2 |x_i|$. Construct a set x_n as follows:

I. Choose two lines L_1 and L_2 which divide the plane into four quadrants.

II. Choose four sets A, B, C and D such that each is a copy of $x_{\frac{n-2}{4}}$ and all intersections between lines in A occur within the first quadrant formed by L_1 and L_2, all B intersections in the second, ..., all D intersections in the fourth.

This yields the structure



Note that we put no restrictions on the locations of intersections of lines from different sets.

Now, we may search this set by first determining in which quadrant the point to be searched for lies. This requires 2 comparisons. Assume without loss of generality that the point lies in quadrant 1. We then consider the complexity of searching AuBuCuD in the first quadrant. However,

$$c_1 (A \cup B \cup C \cup D) \leq c_1(A) + c_1(B) + c_1(c) + c_1(D)$$

where c_1 represents the complexity of searching in the first quadrant. We observe that $c_1(B)$, $c_1(C)$ and $c_1(D)$ are at most $\log_2(\frac{n-2}{4})$ as the lines in each of these sets have no intersections in the first quadrant and hence are totally ordered here. By induction, $c_1(A) \leq k \log^2(\frac{n-2}{4})$. Hence

$$c(x_n) \leq 2 + 3 \log_2(\tfrac{n-2}{4}) + k \log^2(\tfrac{n-2}{4}) \leq$$

$$k \log^2 n + (3-4k) \log n + (4k-4) \leq k \log^2 n \qquad \text{for } k \geq 3/4$$

Q. E. D.

By methods of Lipton-Dobkin [6] we can use theorem 3 to demonstrate that there is a hierarchy in the following sense:

Corollary 4: For any monotone f(n) such that $f(n) \leq n$ and $\frac{f(n)}{\log^2 n} \to \infty$ there is a family $\{A_n\}$ such that $|A_n| = n$ and

$$f(n) \leq c(A_n) \leq O(f(n)).$$

As stated earlier we will now show that c(A) behaves poorly with respect to disjoint union.

Theorem 5: For any $n \geq 1$ there are $|A| = |B| = n$ sets of lines in general position such that $A \cup B$ is also in general position and

(1) $c(A \cup B) = O(\log^2 n)$

(2) $c(A) + c(B) \geq cn$ for some constant $c > 0$.

Sketch of Proof: Let A be a set of n lines in general position such that $r(A) = n$; let R be this region with n sides. Let B be the set of m lines constructed in theorem 3 positioned so that all the intersection points formed by the lines of B lie within R. Now to search $A \cup B$ we proceed as follows: First, determine where with respect to B we are. This can be done in $O(\log^2 m)$ steps. Second, if we are in a bounded region of B, then we must lie inside R and we are done. On the other hand, if we are in an unbounded region we argue as follows. The m+1 unbounded regions of B can be arranged with respect to A so that we can determine where we are in at most $O(\log^2 n)$ additional steps.

References

[1] D. Dobkin. A non-linear lower bound on linear search tree programs for solving knapsack problems. To appear.

[2] D. Dobkin and R. Lipton. On some generalizations of binary search. ACM Symposium on the Theory of Computing, Seattle, Washington, May 1974. (To appear in SIAM Journal of Computing.)

[3] D. Dobkin and R. Lipton. On the complexity of computations under varying sets of primitive operations. Automata Theory and Formal Languages, 2nd GI Conference, Spring-Verlag Lecture Notes in Computer Science, #33.

[4] D. Dobkin and R. Lipton. A lower bound of n^2 on the knapsack problem. In preparation.

[5] D. Knuth. The Art of Computer Programming: Sorting and Searching, III. Reading, Massachusetts: Addison Wesley, 1973.

[6] R. Lipton and D. Dobkin. Complexity measures and hierarchies for the evaluation of integers, polynomials and n-linear forms. ACM Symposium on the Theory of Computing, Albuquerque, New Mexico, May 1975.

[7] M. Rabin. Proving simultaneous positivity of linear forms. JCSS 6: 639-650, 1972.

[8] C. P. Schnorr. The network-complexity of equivalence and other applications of the network complexity. Automata Theory and Formal Languages, 2nd GI Conference, Spring-Verlag Lecture Notes in Computer Science, #33.

[9] M. Shamos. Geometric Complexity. PhD thesis, Yale University, New Haven, Connecticut. To appear.

[10] P. Spira. Complete linear proofs of systems of linear inequalities. JCSS 6:205-216, 1972.

[11] A. Yao. On the complexity of comparison problems using linear functions. IEEE 16th Annual Symposium on Foundations of Computer Science, Berkeley, California, October 1975.