Four lists each containing at MOST 100 docIDs ranked by score with text snippet

Output file: searchResult.txt

a) **Query 1:** portable operating systems

Query id	Rank	Doc ID	Score	Text Snippet
1	1	3127	1.1755296	Thoth, a <B>Portable</B> Real-Time <B>Operating</B> System Thoth isa real-time <B>operating</B> system which is designed to be <B>portable</B> over a large set of machines. It is currently running on two minicomputers with
1	2	1461	0.50642985	Discussion Summary on <B>Operating</B> <B>Systems</B> CACM March, 1966 CA660311 JB March 3, 1978 1:26 PM
1	3	3068	0.40902123	A Model for Verification of Data Security in <B>Operating</B> <B>Systems</B> Program verification applied to kernel architectures forms a promising method for providing uncircumventably secure, shared computer
1	4	2246	0.3299833	Levels of Language for <B>Portable</B> Software An increasing amount of software is being implemented in a <B>portable</B> form. A popular way of accomplishing this is to encode the software in a specially
1	5	2111	0.3230007	Spelling Correction in <B>Systems</B> Programs Several specialized techniques are shown for efficiently incorporating spelling correction algorithms in to compilers and <B>operating</B> <B>systems</B>. These include
1	6	2069	0.31651866	Comments on a Paper by Wallace and Mason CACM April, 1970 Heess Jr., W. F. page-on-demand, demand paging, time-sharing multiprogramming, Markovian computer models, scheduling strategies, <B>operating</B>
1	7	2319	0.30766326	increasingly more common and necessary in the near future. Such a design philosophy will clearly have a severe impact on the way we go about modularizing <B>operating</B> and computer <B>systems</B>. CACM July
1	8	2740	0.30766326	free of "deadly embrace" (deadlock). The design principle is an alternative to Dijkstra's hierarchical structuring of <B>operating</B> <B>systems</B>. The project management and the performance are discussed, too

1	9	1462	0.28890023	Multilevel <B>Operating</B> <B>Systems</B> The Basic software for all newer computers is built on the well-established need for standard <B>operating</B> <B>systems</B>. This implies that all applications-no matter how large
1	10	1728	0.28890023	<B>systems</B>, paging, dynamic program behavior, program behavior, virtual memory <B>systems</B>, single-level storage, one-level storage, <B>operating</B> system simulation, <B>operating</B> <B>systems</B>, supervisor simulation, machine
1	11	2597	0.28045723	buffer pool, a disk head optimizer, and a version of the problem of readers and writers. CACM October, 1974 Hoare, C. A. R. monitors, <B>operating</B> <B>systems</B>, scheduling, mutual exclusion
1	12	2629	0.28045723	features seldom found even in larger <B>operating</B> <B>systems</B>, including: (1) a hierarchical file system incorporating demountable volumes; (2) compatible file, device, and inter-process I/O; (3) the ability to
1	13	1750	0.2741132	place of the faster memory. CACM May, 1968 Fuchel, K. Heller, S. multiple computer <B>systems</B>, extended core storage, multiprogrammed <B>operating</B> <B>systems</B>, multiprocessor <B>operating</B> <B>systems</B>, control
1	14	2372	0.26916727	several existing <B>systems</B>, and serves as a framework for a proposal for general security system implementation within today's languages and <B>operating</B> <B>systems</B>. CACM April, 1972 Conway, R. W. Maxwell, W
1	15	1247	0.26857498	An <B>Operating</B> Environment for Dynamic-Recursive Computer Programming <B>Systems</B> Presented in this paper is a brief nontechnical introduction to OEDIPUS, a computer programming system which can serve as
1	16	2080	0.26857498	The Nucleus of a Multiprogramming System This paper describes the philosophy and structure of a multiprogramming system that can be extended with a hierarchy of <B>operating</B> <B>systems</B> to suit diverse
1	17	2482	0.26857498	, resource allocation, <B>operating</B> <B>systems</B>, multiprogramming, hierarchical <B>systems</B> 4.30 4.32 CA730704 JB January 23, 1978 12:46 PM

1	18	2867	0.26857498	Modularization and Hierarchy in a Family of <B>Operating</B> <B>Systems</B> This paper describes the design philosophy used in the construction of a family of <B>operating</B> <B>systems</B>. It is shown that the concepts of
1	19	2379	0.26133278	on the principles which guided the design. CACM March, 1972 Liskov, B. H. <B>operating</B> <B>systems</B>, system design, levels of abstraction, machine architecture, microprogramming, segments, semaphores
1	20	1923	0.25321493	methods for programs to communicate with deeply embedded facilities such as command language processors. CACM March, 1969 Rosin, R. F. <B>operating</B> <B>systems</B>, interfaces input-output, high level
1	21	1680	0.24681911	retrieval <B>operating</B> <B>systems</B>, graphics, displays, man-machine interface, on-line computing, graphic programming 1.5 3.3 3.5 3.8 4.0 4.3 CA681007 JB February 21, 1978 3:36 PM
1	22	2840	0.24135262	Protection in <B>Operating</B> <B>Systems</B> A model of protection mechanisms in computing <B>systems</B> is presented and its appropriateness is argued. The "safety" problem for protection <B>systems</B> under this model
1	23	3025	0.24135262	<B>systems</B>. CACM January, 1978 Bell, C. Kotok, A. Hastings, T. Hill, R. computer structures, architecture, <B>operating</B> system, timesharing 4.32 6.21 6.3 CA780105 JB March 28, 1978 5:31 PM
1	24	2621	0.23838985	superior to one-way ciphers derived from Sannon codes. CACM August, 1974 Purdy, G. B. <B>operating</B> <B>systems</B>, time sharing <B>systems</B>, security, cryptography 4.35 CA740803 JB January 17, 1978 10:10 AM
1	25	2378	0.23371433	concepts in detail, outlines the remainder of the supervisor, and discusses some of the advantages of this approach. CACM March, 1972 Gaines, R. S. <B>operating</B> <B>systems</B>, supervisors, multiprogramming, time
1	26	2541	0.23371433	subsystems. It also suggested an important new concept for <B>operating</B> <B>systems</B>: separation of the scheduling from the maintenance functions in resource allocation. This separation enables incorporation of
1	27	2632	0.23371433	secure <B>systems</B>. CACM June, 1974 Wulf, W. Cohen, E. Corwin, W. Jones, A. Levin, R. Pierson, C. Pollack, F. <B>operating</B> system, kernel, nucleus, protection, security 4.3 6.2 CA740614 JB January

1	28	1747	0.23342077	design, <B>operating</B> <B>systems</B>, <B>operating</B> <B>systems</B> design, multiprogramming, multiprogrammed <B>systems</B>, multiprogrammed system design, virtual computers, programming languages, programming language design
1	29	2317	0.23112017	microprogramming for some programming system functions. CACM July, 1972 Rosen, S. languages, <B>operating</B> <B>systems</B>, programming <B>systems</B>, multiprogramming, history 1.2 4.22 4.32 CA720711 JB January
1	30	2424	0.22735322	single hardware or software fault.the amount of additional hardware and software required for dynamic verification can be modest. CACM November, 1973 Fabry, R. S. <B>operating</B> <B>systems</B>, data security
1	31	2138	0.22647284	machine (the PDP-10): compilers, <B>operating</B> <B>systems</B>, etc. Prime design goals of the design are the ability to produce highly efficient object code, to allow access to all relevant hardware features
1	32	2151	0.22647284	. <B>operating</B> <B>systems</B>, multiprogramming <B>systems</B>, time-sharing <B>systems</B>, software measurement, user program measurement, measurement technology, TX-2 computer, virtual computers, performance improvement
1	33	1751	0.2238125	, multiprocessing, multiprogramming, <B>operating</B> <B>systems</B>, program behavior, program models, resource allocation, scheduling, storage allocation 4.30 4.32 CA680505 JB February 23, 1978 9:33 AM
1	34	2320	0.2238125	within <B>operating</B> <B>systems</B> but also within user programs. CACM July, 1972 Hansen, P. B. structured multiprogramming, programming languages, <B>operating</B> <B>systems</B>, concurrent processes, shared data
1	35	2950	0.2238125	framework for implementing flexible schedulers in real <B>operating</B> <B>systems</B>. The policy-driven scheduler of Bernstein and Sharp is discussed as an example of such an implementation CACM July
1	36	3196	0.22226858	preferred. The reactive typewriter should be <B>portable</B>. the reactive typewriter should operate over any commercially used, dial-type telephone (voice) or telegraph (Telex) line or over leased (nondial
1	37	3174	0.22156307	countering observed attempts to penetrate the system. The result is a compromise between extreme security and ease of use. CACM November, 1979 Morris, R. Thompson, K. <B>Operating</B> <B>systems</B>, passwords

1	38	1930	0.21998885	Extremely <B>Portable</B> Random Number Generator Extremely <B>portable</B> subroutines are sometimes needed for which moderate quality and efficiency suffice. Typically, this occurs for library functions (like
1	39	1226	0.21734169	the system's <B>operating</B> efficiency. This paper provides the <B>systems</B> designer with an information source which describes ten techniques that may be employed for organizing structured data. The
1	40	3141	0.21734169	<B>systems</B>, <B>operating</B> <B>systems</B> 4.32 4.35 5.25 5.32 CA790502 DH June 5, 1979 2:35 PM
1	41	1324	0.21344522	Answering English questions by Computer: A Survey Fifteen experimental English language question-answering <B>systems</B> which are programmed and <B>operating</B> are described and reviewed. The <B>systems</B> range
1	42	1951	0.21344522	distribution by the exponential distribution are discussed for the <B>systems</B> studied. CACM December 1970 Fuchs, E. Jackson, P. E. computer communications, time-sharing, <B>operating</B> <B>systems</B>, optimization
1	43	2522	0.21344522	April, 1973 Rodriguez-Rosell, J. Dupuy, J. P. working set, dispatcher, scheduler, time-sharing <B>systems</B>, resource allocation, software evaluation, <B>operating</B> <B>systems</B>, supervisory <B>systems</B> 4.31 4.32 CA
1	44	1236	0.20095375	The SMART Automatic Document Retrieval System-An Illustration A fully automatic document retrieval system <B>operating</B> on the IBM 7094 is described. The system is characterized by the fact that
1	45	1752	0.20095375	. Weizer, N. time-sharing, <B>operating</B> <B>systems</B>, resource management, task scheduling, paging, system simulation, memory management, virtual memories 4.30 4.31 4.32 CA680504 JB February
1	46	2777	0.20095375	solution to the problem. It also discusses the need for the generalized operators suggested by Patil. CACM March, 1975 Parnas, D. L. <B>operating</B> <B>systems</B>, co-<B>operating</B> processes,process synchronization
1	47	322	0.1986582	Operational Compatibility of <B>Systems</B>-CONVENTIONS The General Standards Committee of the SHARE organization has devoted considerable effort to the problem of <B>operating</B> a computer efficiently in view

1	48	2624	0.1986582	Formal Requirements for Virtualizable Third Generation Architectures Virtual machine <B>systems</B> have been implemented on a limited number of third generation computer <B>systems</B>, e.g. CP-67 on the IBM
1	49	3026	0.1986582	The Evolution of the Sperry Univac 1100 Series: A His tory, Analysis, and Projection The 1100 series <B>systems</B> are Sperry Univac's large-scale main frame computer <B>systems</B>. Beginning with the 1107 in
1	50	2358	0.1930821	The Multics Virtual Memory: Concepts and Design As experience with use of on-line <B>operating</B> <B>systems</B> has grown, the need to share information among system users has become increasingly apparent
1	51	2374	0.18991122	A Study of Storage Partitioning Using a Mathematical Model of Locality Both fixed and dynamic storage partitioning procedures are examined for use in multiprogramming <B>systems</B>. The storage
1	52	3137	0.18991122	A Methodology for the Design of Distributed Information <B>Systems</B> A macro model of a distributed information system in presented. The model describes the major costs of using an information system
1	53	2036	0.18111807	-teletype, interaction, conditional job control, <B>operating</B> <B>systems</B> 3.80 3.81 4.29 4.39 CA700701 JB February 13, 1978 9:43 AM
1	54	3028	0.18111807	performance of the Mark I and Atlas is evaluated. CACM January, 1978 Lavington, S. architecture, index registers, paging, virtual storage, extra codes, compilers, <B>operating</B> <B>systems</B>, Ferranti, Manchester
1	55	2184	0.160763	On the Meaning of Names in Programming <B>Systems</B> It is assumed that there is a similarity of function between the data names of a programming language and the file names of an <B>operating</B> system. The
1	56	2297	0.160763	. CACM August, 1972 Oden, P. H. Shedler, G. S. paging machines, demand paging, <B>operating</B> <B>systems</B> studies, queuing analysis, memory contention, memory management 4.32 CA720805 JB January
1	57	2622	0.160763	A User Authentication Scheme Not Requiring Secrecy in the Computer In many computer <B>operating</B> <B>systems</B> a user authenticates himself by entering a secret password known solely to himself and the

1	58	1892	0.15892656	multiprogramming, Markovian computer models, scheduling strategies, <B>operating</B> <B>systems</B>, memory management 2.44 4.32 4.39 6.20 6.21 CA690601 JB February 17, 1978 1:43 PM
1	59	2625	0.15892656	, protection hardware, shared addresses, information sharing, <B>operating</B> <B>systems</B>, computer utility, segmentation, tagged architecture 4.30 4.32 4.34 6.21 CA740706 JB January 17, 1978 12:39 PM
1	60	696	0.15825933	Company's manufacturing facilities located in Van Nuys and Sunnyvale, California. The system includes over 200 remote Input Stations which collect and transmit Company <B>operating</B> data to a central Data
1	61	1647	0.15825933	WATFOR-The University of Waterloo FORTRAN IV Compiler WATFOR is an in-core, load-and-go compiler which has been implemented within the IBM 7040/44 <B>operating</B> system. FORTRAN IV was selected as the
1	62	2095	0.15825933	Measurements of Segment Size Distributions of segment sizes measured under routine <B>operating</B> con ditions on a computer system which utilizes variable sized segments (the Burroughs B5500) are
1	63	2582	0.15825933	Improving Locality by Critical Working Sets A new approach to program locality improvement via restructuring is described. The method is particularly suited to those <B>systems</B> where primary memory
1	64	2972	0.15825933	large programs, such as <B>operating</B> <B>systems</B>. CACM April, 1977 Robinson, L. Levitt, K. N. hierarchical structure, program verification, structured programming, formal specification, abstraction, and
1	65	3006	0.15825933	Anomalies with Variable Partition Paging Algorithms Five types of anomalous behavior which may occur in paged virtual memory <B>operating</B> <B>systems</B> a redefined. One type of anomaly, for example
1	66	3050	0.15825933	<B>Systems</B> Design Education: A Gaming Approach One of the problems facing managers of computer installations is the problem of configuring the computer system to meet the demands made by the mix of
1	67	3105	0.15825933	A Language Extension for Expressing Constraints on Data Access Controlled sharing of information is needed and desirable for many applications and is supported in <B>operating</B> <B>systems</B> by access
1	68	1069	0.14430283	A Method for Comparing the Internal <B>Operating</B> Speeds of Computers CACM May, 1964 Raichelson, E. Collins, G. CA640520 JB March 9, 1978 11:35 PM

1	69	1755	0.14430283	Proceedings of the ACM Symposium on <B>Operating</B> system Principles CACM May, 1968 ACM Special Interest Committee CA680501 JB February 23, 1978 9:56 AM
1	70	2796	0.14430283	Monitors: An <B>Operating</B> System Structuring Concept (Corrigendum) CACM February, 1975 Hoare, C. A. R. CA750203 JB January 12, 1978 8:45 AM
1	71	1517	0.12660746	Methods for Analyzing Data from Computer Simulation Experiments This paper addresses itself to the problem of analyzing data generated by computer simulations of economic <B>systems</B>. We first turn to
1	72	1810	0.12660746	folding, are discussed, and it is shown that given some care in use the unit performs satisfactorily under the conditions tested, even though it is <B>operating</B> across a memory-to-storage interface
1	73	1854	0.12660746	specification of concurrent (or pseudoconcurrent) activities in a supposedly more perspicuous manner. It is intended to serve as a basis for the construction of <B>operating</B> <B>systems</B>, which are prime
1	74	1877	0.12660746	Prevention of System Deadlocks A well-known problem in the design of <B>operating</B> <B>systems</B> is the selection of a resource allocation policy that will prevent deadlock. Deadlock is the situation in
1	75	2342	0.12660746	. CACM June, 1972 Gilbert, P. Chandler, W. J. concurrent programming control, cooperating processes, formal programs, interference, mutual exclusion, <B>operating</B> <B>systems</B>, parallel processes
1	76	2535	0.12660746	conditions under which this output process is approximately Poisson. CACM March, 1973 Pack, C. D. computer communications, time-sharing, multiplexing, scheduling algorithms, <B>operating</B> <B>systems</B> 3.80 3.81 6.20 CA730304 JB January 24, 1978 11:13 AM
1	77	2542	0.12660746	. DeVaney, D. B. <B>operating</B> system development, language processing software evaluation, compilers, system programming, supervisory <B>systems</B>, debugging, program maintenance, modeling, system integration
1	78	2849	0.12660746	stations. The packet transport mechanism provided by Ethernet has been used to build <B>systems</B> which can be viewed as either local computer networks or loosely coupled multiprocessors. An Ethernet's
1	79	3002	0.12660746	<B>systems</B> may be derived from it by an appropriate selection of its parameters. This model has already been used in the optimization of library routines' storage at a large scale <B>operating</B> system

1	80	1050	0.10307345	A Parts Breakdown Technique Using List Structures List structured parts breakdown is proposed and discussed. Implementation facts are presented on <B>operating</B> program using these techniques. CACM
1	81	1472	0.10203751	Description of a High Capacity, Fast Turnaround University Computing Center The <B>operating</B> system for the UNIVAC 1107 at Case Institute is reviewed. The system is of interest because of the low
1	82	1591	0.092191696	A Model for a Multifunctional Teaching System A teaching system model that was incorporated into an <B>operating</B> system of a large computer is described. The model transferred control to the
1	83	2357	0.08926423	MUX, a Simple Approach to On-Line Computing An on-line system <B>operating</B> as part of a normal batch system for the CDC 6600 computer is described. The system, which required one man-year for initial
1	84	2868	0.08926423	Reflections on an <B>Operating</B> System Design The main features of a general purpose multiaccess <B>operating</B> system developed for the CDC 6400 at Berkeley are presented, and its good and bad points are
1	85	2920	0.08926423	, <B>operating</B> system, etc., of the "safe situations" which may be realized without endangering the smooth running of the system. When each process specifies its future needs by a flowchart of need
1	86	18	0.08829741	Simple Automatic Coding <B>Systems</B> CACM July, 1958 Adams, E. S. Schlesinger, S. I. CA580701 JB March 22, 1978 9:06 PM
1	87	25	0.08829741	General Purpose Programming <B>Systems</B> CACM May, 1958 Holt, A. W. CA580503 JB March 22, 1978 9:14 PM
1	88	597	0.08829741	Modular Data Processing <B>Systems</B> Written in COBOL CACM May, 1962 Emery, J. C. CA620508 JB March 20, 1978 1:47 PM
1	89	652	0.08829741	Use of Semantic Structure in Information <B>Systems</B> CACM January, 1962 Sable, J. D. CA620107 JB March 20, 1978 4:37 PM
1	90	1360	0.08829741	Description of <B>Systems</B> Used for Data Transmission* (An ASA Tutorial) CACM October, 1966 CA661007 JB March 2, 1978 3:54 PM
1	91	1464	0.08746072	An ALGOL Compiler: Construction and Use in Relation to an Elaborate <B>Operating</B> System An ALGOL translator has been prepared and integrated into the IBSYS <B>Operating</B> System. Assembly and "go

1	92	2311	0.08637957	processing, hardware <B>systems</B>, software <B>systems</B>, <B>systems</B> development process, functional <B>systems</B>, management <B>systems</B> 2.0 3.50 4.0 6.20 CA720717 JB January 30, 1978 11:22 AM
1	93	2407	0.08637957	Curriculum Recommendations for Undergraduate Programs in Information <B>Systems</B> The need for education related to information <B>systems</B> in organizations is discussed, and a curriculum is proposed for an
1	94	2345	0.08110635	Curriculum Recommendations for Graduate Professional Programs in Information <B>Systems</B> The need for education related to information <B>systems</B> in organizations is discussed, and a curriculum is
1	95	2583	0.07804462	Guidelines for Humanizing Computerized Information <B>Systems</B>: A Report from Stanley House CACM November, 1974 Sterling, T. D. humanization, social implication, management, information <B>systems</B>
1	96	238	0.07726023	Simulation and Analysis of Biochemical <B>Systems</B> CACM December, 1961 Garfinkel, D. Rutledge, J. D. Higgins, J. J. CA611212 JB March 15, 1978 10:09 PM
1	97	651	0.07726023	A Survey of Languages and <B>Systems</B> for Information Retrieval CACM January, 1962 Grems, M. CA620108 JB March 20, 1978 4:36 PM
1	98	795	0.07726023	Debugging <B>Systems</B> at the Source Language Level CACM August, 1963 Ferguson, H. E. Berner, E. CA630803 JB March 13, 1978 8:46 PM
1	99	1912	0.07726023	Simulation of Outpatient Appointment <B>Systems</B> An experimental computer program is described which simulates appointment <B>systems</B> employed by outpatient departments of hospitals. Both major kinds of
1	100	2690	0.07726023	A Numbering <B>Systems</B> for Combinations CACM January, 1974 Knott, G. D. combinatorics, coding system, storage mapping function 4.9 5.30 CA740109 JB January 18, 1978 1:42 PM

b) **Query 2:** code optimization for space efficiency

Query id	Rank	Doc ID	Score	Text Snippet
2	1	1795	0.64332205	Optimal \<B>Code\</B> \<B>for\</B> Serial and Parallel Computation CACM December, 1969 Fateman, R. J. \<B>code\</B> \<B>optimization\</B>, sequencing of operations, detection of common subexpressions 4.12 CA691217 JB February 15, 1978 1:59 PM
2	2	2495	0.5975083	Adapting Optimal \<B>Code\</B> Generation \<B>for\</B> Arithmetic Expressions to the Instruction Sets Available on Present-Day Computers CACM June, 1973 Stockhausen, P. F. arithmetic expressions, \<B>code\</B> generation
2	3	1947	0.5535952	Object \<B>code\</B> \<B>Optimization\</B> Methods of analyzing the control flow and data flow of programs during compilation are applied to transforming the program to improve object time \<B>efficiency\</B>. Dominance
2	4	2748	0.528035	Indirect Threaded \<B>Code\</B> An efficient arrangement \<B>for\</B> interpretive \<B>code\</B> is described. It is related to Bell's notion of threaded \<B>code\</B> but requires less \<B>space\</B> and is more amenable to machine
2	5	2559	0.4950907	The Reallocation of Hash-Coded Tables When the \<B>space\</B> allocation \<B>for\</B> a hash-coded table is altered, the table entries must be rescattered over the new \<B>space\</B>. A technique \<B>for\</B> accomplishing this
2	6	1886	0.44647112	Generation of Optimal \<B>Code\</B> \<B>for\</B> Expressions via Factorization Given a set of expressions which are to be compiled, methods are presented \<B>for\</B> increasing the \<B>efficiency\</B> of the object \<B>code\</B> produced by
2	7	2897	0.43843484	produce good \<B>code\</B>. More elaborate optimizations can further improve the object \<B>code\</B>. \<B>For\</B> most contexts of the concatenate statement, the \<B>code\</B> produced by a compiler using the expansion- \<B>optimization\</B>
2	8	2904	0.38659886	. compilers, \<B>optimization\</B> of compiled \<B>code\</B>, program analysis, operator strength reduction, test replacement, strongly connected region 4.12 5.24 5.32 CA771112 JB December 27, 1977 6:34 AM

2	9	3080	0.38377044	Proving the Correctness of Heuristically Optimized <B>Code</B> A system <B>for</B> proving that programs written in a high level language are correctly translated to a low level language is described. A
2	10	2491	0.37791464	interpretive <B>code</B> not needing an interpreter. Extensions and optimizations are mentioned. CACM June, 1973 Bell, J. R. interpreter, machine <B>code</B>, time tradeoff, <B>space</B> tradeoff, compiled <B>code</B>, subroutine
2	11	2680	0.3617208	suggested <B>for</B> those cases in which it is inefficient to compute the actual optimum. CACM March, 1974 Morgan, H. L. disk analysis, disk <B>optimization</B>, disk files,file systems, file scheduling, <B>space</B>
2	12	2537	0.3568571	Common Phrases and Minimum-<B>Space</B> Text Storage A method <B>for</B> saving storage <B>space</B> <B>for</B> text strings, such as compiler diagnostic messages, is described. The method relies on hand selection of a set
2	13	3171	0.34712863	absolutely no overhead, in either time or <B>space</B>, during execution of the program. CACM October, 1979 Klint, P. Line number administration, diagnostic messages, abstract machine <B>code</B> 4.12 4.13 4.20 4.42 CA791004 DB January 17, 1980 9:57 AM
2	14	2464	0.3463252	The Complex Method <B>for</B> Constrained <B>Optimization</B> [E4] (Algorithm A454) CACM August, 1973 Richardson, J. A. Kuester, J. L. <B>optimization</B>, constrained <B>optimization</B>, Box's algorithm 5.41 CA730810 JB January 23, 1978 10:34 AM
2	15	1195	0.33868977	UPLIFTS-University of Pittsburgh Linear File Tandem System A series of computer programs has been developed and is now operational <B>for</B> processing the National Aeronautics and <B>Space</B> Administration
2	16	2253	0.32735753	calculi described by the authors in a previous paper. This scheme attempts program <B>optimization</B> by transforming the original algorithm rather than the machine <B>code</B>. The goal is to automatically
2	17	2944	0.3262091	that cancel when the <B>code</B> is executed and those that can be grouped to achieve improved <B>efficiency</B>. CACM July, 1977 Barth, J. M. garbage collection, global flow analysis, list processing
2	18	2929	0.31939238	An Analysis of Inline Substitution <B>for</B> a Structured Programming Language An <B>optimization</B> technique known as inline substitution is analyzed. The <B>optimization</B> consists of replacing a procedure

2	19	1655	0.30841085	<B>Code</B> Extension Procedures <B>for</B> Information Interchange* (Proposed USA Standard) CACM December, 1968 standard <B>code</B>, <B>code</B>, information interchange, characters, shift out, shift in, escape, data link
2	20	2716	0.28945735	, which was designed using this approach, is described in detail. SQUIRAL seeks to minimize query response time and <B>space</B> utilization by: (1) performing global query <B>optimization</B>, (2) exploiting disjoint
2	21	1564	0.2743894	Description of Basic Algorithm in DETAB/65 Preprocessor The basic algorithm <B>for</B> the conversion of decision tables into COBOL <B>code</B> is contained in the generator portion of the DETAB/65 preprocessor
2	22	2423	0.2743894	minute. <B>For</B> an XPL compiler, the parser program and its tables currently occupy 288 words of 60-bit core memory of which 140 words are parsing table entries and 82 words are links to <B>code</B>
2	23	2374	0.27298835	A Study of Storage Partitioning Using a Mathematical Model of Locality Both fixed and dynamic storage partitioning procedures are examined <B>for</B> use in multiprogramming systems. The storage
2	24	3005	0.26183242	Implications of Structured Programming <B>for</B> Machine Architecture Based on an empirical study of more than 10,000 lines of program text written in a GOTO-less language, a machine architecture
2	25	2033	0.26124457	<B>Space</B>/Time Trade-offs in Hash Coding with Allowable Errors In this paper trade-offs among certain computational factors a given set of messages. Two new hash-coding methods are examined and
2	26	1651	0.24911706	subsequent use either as a pedagogical device or <B>for</B> solving rather small LP problems. This latter (limited) use derives not at all from inherent limitations in the <B>code</B> itself, but from an <B>efficiency</B>
2	27	2611	0.24534898	The Complex Method <B>for</B> Constrained <B>Optimization</B> (Algorithm R454) CACM August, 1974 Shere, K. D. CA740813 JB January 17, 1978 9:28 AM
2	28	2858	0.24334013	addressing constraints. This may be, <B>for</B> example, achieving the smallest number of long instructions, in which case the total <B>code</B> length is minimized, or minimizing the assigned address of a specified
2	29	3053	0.24334013	) <B>optimization</B>, although packing costs remain linear-O(n)-with table size n. The techniques are primarily suited <B>for</B> important fixed (but possibly quite large) tables <B>for</B> which reference frequencies may

2	30	3054	0.24068223	sorting method which requires negligible extra storage. CACM October, 1978 Sedgewick, R. Quicksort, analysis of algorithms, <B>code</B> <B>optimization</B>, sorting 4.0 4.6 5.25 5.31 5.5 CA781007 DH January
2	31	267	0.24038294	Some Proposals <B>for</B> Improving the <B>Efficiency</B> of ALGOL 60 CACM November, 1961 Wilkes, M. V. CA611101 JB March 15,1978 11:01 PM
2	32	230	0.23872676	A Proposal <B>for</B> Character <B>Code</B> Compatibility CACM February, 1960 Bemer, R. W. CA600202 JB March 22, 1978 2:25 PM
2	33	797	0.23872676	American Standard <B>Code</B> <B>for</B> Information Interchange CACM August, 1963 CA630801 JB March 13, 1978 8:55 PM
2	34	1064	0.23872676	Perforated Tape <B>Code</B> <B>for</B> Information Interchange (Proposed American Standard) CACM June, 1964 CA640603 JB March 9, 1978 9:46 PM
2	35	1289	0.23872676	Proposed Revised American Standard <B>Code</B> <B>for</B> Information Interchange CACM April, 1965 CA650402 JB March 7, 1978 4:03 PM
2	36	2856	0.23869252	<B>space</B>, <B>for</B> example) is located only through implicit enumeration of all possible decision trees using a technique such as branch-and-bound. The new approach described in this paper uses dynamic
2	37	2433	0.23437612	that data allocation rather than <B>code</B> structuring is the crucial ILLIAC <B>optimization</B> problem. A satisfactory method of data allocation is then presented. Language structures to utilize this storage
2	38	1234	0.23171572	-language. A technique of simulation at compile time of the use of a conventional run-time stack enables the generation of <B>code</B> <B>for</B> expressions which minimizes stores, fetches and stack-pointer
2	39	2701	0.22340249	December, 1975 Graham, S. L. Wegman, M. global flow analysis, data flow, <B>code</B> <B>optimization</B>, common subexpression elimination, live-dead analysis, information propagation, flow graph, reducibility
2	40	1901	0.22280307	Dynamic <B>Space</B>-Sharing in Computer Systems A formalization of relationships between <B>space</B>-shading program behavior, and processor <B>efficiency</B> in computer systems is presented. Concepts of value and
2	41	66	0.22071782	A Proposal <B>for</B> a Generalized Card <B>Code</B> <B>for</B> 256 Characters CACM September, 1959 Bemer, R. W. CA590903 JB March 22, 1978 4:50 PM

2	42	1757	0.21100664	Data <B>Code</B> <B>for</B> Calendar Date <B>for</B> Machine-to-Machine Data Interchange* (Proposed USA Standard) CACM April, 1968 USA Standard, data <B>code</B>, calendar date, machine-to-machine data interchange, recording
2	43	1542	0.2106169	A Microprogrammed Implementation of EULER on IBM System/360 Model 30 An experimental processing system <B>for</B> the algorithmic language EULER has been implemented in microprogramming on an IBM System
2	44	2133	0.21029912	Algorithm <B>for</B> the Assignment Problem (Rectangular Matrices) [H] (Algorithm 415) CACM December, 1971 Bourgeois, F. Lassalle, J. C. operations research, <B>optimization</B> theory, assignment problem
2	45	1275	0.20888591	<B>Code</B> Structures <B>for</B> Protection and Manipulation of Variable Length Items (Corrigendum) CACM April, 1965 Ramamoorthy, C. V. CA650416 JB March 7, 1978 3:31 PM
2	46	1414	0.20888591	Twelve-Row Punched-Card <B>Code</B> <B>for</B> Information Interchange* (Proposed American Standard) CACM June, 1966 CA660607 JB March 3, 1978 8:39 AM
2	47	2835	0.20701885	Recursion Analysis <B>for</B> Compiler <B>Optimization</B> A relatively simple method <B>for</B> the detection of recursive use of procedures is presented <B>for</B> use in compiler <B>optimization</B>. Implementation
2	48	149	0.20604253	A Decision Rule <B>for</B> Improved <B>Efficiency</B> in Solving Linear Programming Problems with the Simplex Algorithm CACM September, 1960 Dickson, J. C. Frederick, F. P. CA600908 JB March 20, 1978 8:43 PM
2	49	1670	0.20255528	Correspondences of 8-Bit and Hollerith Codes <B>for</B> Computer Environments (A USASI Tutorial Standard) CACM November, 1968 USA standard, card <B>code</B>, punched card, punched card <B>code</B>, hole-patterns, hole
2	50	1992	0.20255528	Comment on Bell's Quadratic Quotient Method <B>for</B> Hash <B>Code</B> Searching CACM September, 1970 Lamport, L. hashing, hash <B>code</B>, scatter storage, calculated address, clustering, search, symbol table, keys
2	51	2722	0.19925252	Multidimensional Binary Search Trees Used <B>for</B> Associative Searching This paper develops the multidimensional binary search tree (or k-d tree, where k is the dimensionality of the search <B>space</B>) as a

2	52	2745	0.19869836	A Linear <B>Space</B> Algorithm <B>for</B> Computing Maximal Common Subsequences The problem of finding a longest common subsequence of two strings has been solved in quadratic time and <B>space</B>. An algorithm is
2	53	2734	0.19712423	, the size of the request population had little effect on allocation <B>efficiency</B>. <B>For</B> exponential and hyperexponential distributions of requests, first-fit outperformed best-fit; but <B>for</B> normal and
2	54	2530	0.18242602	An Algorithm <B>for</B> Extracting Phrases in a <B>Space</B>-Optimal Fashion [Z] (Algorithm A444) CACM March, 1973 Wagner, R. A. information retrieval, coding, text compression 3.70 5.6 CA730309 JB January
2	55	3129	0.17914577	Optimal Storage Allocation <B>for</B> Serial Files A computer system uses several serial files. The files reside on a direct-access storage device in which storage <B>space</B> is limited. Records are added
2	56	1065	0.17904507	Bit Sequencing of the American Standard <B>Code</B> <B>for</B> Information Interchange (ASCII) in Serial-by-Bit Data Transmission (Proposed American Standard) CACM June, 1964 CA640602 JB March 9, 1978 9:47 PM
2	57	2586	0.17904507	Adapting Optimal <B>Code</B> Generation <B>for</B> Arithmetic Expressions to the Instruction Sets Available on Present-Day Computers (Errata) CACM October, 1974 Stockhausen, P. F. CA741012 JB January 16, 1978 11:05 AM
2	58	48	0.17705749	Shift-Register <B>Code</B> <B>for</B> Indexing Applications In this communication the use of a shift-register <B>code</B> with n = 10 is described <B>for</B> calling 64 wireless telemetering stations in a fixed cyclical order
2	59	1362	0.17705749	<B>Code</B> Extension in ASCII* (An ASA Tutorial) The American Standard <B>Code</B> <B>for</B> Information Interchange (ASCII) contains a number of control characters associated with the principle of <B>code</B> extension
2	60	2711	0.1756262	A Vector <B>Space</B> Model <B>for</B> Automatic Indexing In a document retrieval, or other pattern matching environment where stored entities (documents) are compared with each other or with incoming patterns
2	61	2078	0.1752548	Representations <B>for</B> <B>Space</B> Planning Problems involving the arrangement of objects in two- or three-<B>space</B> where the objective function primarily consists of derivatives of the distance between

2	62	1807	0.17389299	to produce excellent object <B>code</B> without significantly reducing the compilation speed. CACM December, 1969 Busam, V. A. England, D. E. FORTRAN, <B>optimization</B>, expressions, compilers, compilation
2	63	1223	0.16849889	subexpression recognition. <B>Optimization</B> such as the effective use of index registers, although as important, is not discussed since the object <B>code</B> which would be most efficient is highly machine
2	64	1235	0.16849351	at relatively high speed in only a limited storage <B>space</B>. About half of the word-events in a corpus are identified through the use of a small dictionary of function words and frequently occurring
2	65	3083	0.16805974	fact that link fields are present in each cell of the hash table which permits "chaining" of the first overflow items in the table. The <B>efficiency</B> of the method is derived and a tradeoff analysis is
2	66	1369	0.16523181	Half Rotations in N-Dimensional Euclidean <B>Space</B> An iterative procedure is described <B>for</B> determining half rotations in n-dimensional Euclidean <B>space</B>. The method is a variant of the cyclic Jacobi
2	67	1204	0.15765558	Character Structure and Character Parity Sense <B>for</B> Serial-by-Bit Data Communication in the American Standard <B>Code</B> <B>for</B> Information Interchange (Proposed American Standard) CACM September, 1965 CA650902 JB March 6, 1978 7:41 PM
2	68	2645	0.15531684	Two Languages <B>for</B> Estimating Program <B>Efficiency</B> Two languages enabling their users to estimate the <B>efficiency</B> of computer programs are presented. The program whose <B>efficiency</B> one wishes to
2	69	2892	0.15143192	of protocol hierarchy and the characteristics of each level are summarized. Then the line <B>efficiency</B> <B>for</B> various models of system use is studied. Some measurements of line <B>efficiency</B> <B>for</B> the
2	70	1708	0.14958504	A Note on the <B>Efficiency</B> of a LISP Computation in a Paged Machine The problem of the use of two levels of storage <B>for</B> programs is explored in the context of a LISP system which uses core memory as
2	71	93	0.14920424	From Formulas to Computer Oriented Language A technique is shown <B>for</B> enabling a computer to translate simple algebraic formulas into a three address computer <B>code</B>. CACM March, 1959 Wegstein, J. H

2	72	2389	0.14902377	Preliminary Report on a System <B>for</B> General <B>Space</B> Planning A computer language and a set of programs within that language are described which allow the formulating and solving of a class of <B>space</B>
2	73	2863	0.14902377	VMIN-An Optimal Variable-<B>Space</B> Page Replacement Algorithm A criterion <B>for</B> comparing variable <B>space</B> page replacement algorithms is presented. An optimum page replacement algorithm, called VMIN, is
2	74	1652	0.14447135	A <B>Code</B> <B>for</B> Non-numeric Information Processing Applications in Online Systems A <B>code</B> has been specifically designed to simplify the internal information processing operations within an online
2	75	1676	0.14430216	must be developed, written in a high level language, which minimizes machine dependencies and isolates those which are necessary. A language and a compiler <B>for</B> the language are discussed here. The
2	76	2836	0.14363311	Weighted Derivation Trees The nodes of a weighted derivation tree are associated with weighting functions over the vocabulary of a context-free grammar. An algorithm is presented <B>for</B> constructing
2	77	2524	0.1414585	metric on the the key <B>space</B>, is suitably defined, three file structures are presented together with their corresponding search algorithms, which are intended to reduce the number of comparisons
2	78	1465	0.14062326	Program Translation Viewed as a General Data Processing Problem <B>Efficiency</B> dictates that the overall effectiveness of a compiler be increased by all means available. <B>For</B> a compiler to have a
2	79	2126	0.13736169	Experience with an Extensible Language An operational extensible language system is described. The system and its base language are appraised with respect to <B>efficiency</B>, flexibility, and utility
2	80	1523	0.13666514	SHARER, a Time Sharing System <B>for</B> the CDC 6600 A time sharing system embedded within the standard batch processing system <B>for</B> the CDC 6600 is described. The system is general purpose and file
2	81	2230	0.13616589	A Language <B>for</B> Treating Geometric Patterns in a Two-dimensional <B>space</B> In this paper CADEP, a problem-oriented language <B>for</B> positioning geometric patterns in a two-dimensional <B>space</B>, is presented

2	82	1331	0.13049494	<B>Code</B> Structures <B>for</B> Protection and Manipulation of Variable-Length Items When items are made up of a variable number of characters, each containing the same number of bits, certain control
2	83	2498	0.12899466	Minimizing Wasted <B>Space</B> in Partitioned Segmentation A paged virtual memory system using a finite number of page sizes is considered. Two algorithms <B>for</B> assigning pages to segments are discussed
2	84	794	0.12837844	A List-Type Storage Technique <B>for</B> Alphameric Information A method which is economic in terms of <B>space</B> and time is proposed <B>for</B> the storage and manipulation of character strings of arbitrary length
2	85	2344	0.12832725	On the <B>Optimization</B> of Performance of Time-Sharing Systems by Simulation A simulation model of a time-sharing system with a finite noncontiguous store and an infinite auxiliary store is used to
2	86	2950	0.12738775	. Various classes of scheduling algorithms are defined and related to existing algorithms. A criterion <B>for</B> the implementation <B>efficiency</B> of an algorithm is developed and results in the definition of time
2	87	124	0.12612447	An Algorithm <B>for</B> the Assignment Problem The assignment problem is formulated and briefly discussed. An efficient algorithm <B>for</B> its solution is presented in ALGOL <B>code</B>. An empirical relation
2	88	1551	0.12612447	On Compiling Algorithms <B>for</B> Arithmetic Expressions This paper deals with algorithms concerning arithmetic expressions used in a FORTRAN IV compiler <B>for</B> a HITAC-5020 computer having n accumulators
2	89	1270	0.12610742	On ALGOL Education: Automatic Grading Programs Two ALGOL grader programs are presented <B>for</B> the computer evaluation of student ALGOL programs. One is <B>for</B> a beginner's program; it furnishes random
2	90	1352	0.12610742	Automatic Integration of a Function with a Parameter Two efficient methods <B>for</B> automatic numerical integration are Romberg integration and adaptive Simpson integration. <B>For</B> integrands of the form
2	91	1429	0.12610742	Matrix Reduction Using the Hungarian Method <B>For</B> The Generation of School Timetables The application of Kuhn's Hungarian Method to the problem of matrix reduction as needed in Gotlieb's method <B>for</B>

2	92	2365	0.12610742	Matrix Computations with Fortran and Paging The <B>efficiency</B> of conventional Fortran programs <B>for</B> matrix computations can often be improved by reversing the order of nested loops. Such modifications
2	93	2816	0.12418647	the optimum points at which to reorganize a database. A disk file organization which allows <B>for</B> distributed free <B>space</B> is described. A cost function describing the excess costs due to physical
2	94	2249	0.12392385	A Comparison of Multivariate Normal Generators Three methods <B>for</B> generating outcomes on multivariate normal random vectors with a specified variance-covariance matrix are presented. A comparison
2	95	1030	0.1231686	<B>efficiency</B> of a simulation system, methods are presented <B>for</B> simultaneously evaluating many functions <B>for</B> one set of values of the variables,and <B>for</B> evaluating simultaneously one function <B>for</B> many
2	96	693	0.12267449	An Extension of Fibonaccian Search To Several Variables A technique which uses Fibonaccian search concepts has been developed to solve <B>optimization</B> problems involving unimodal functions of several
2	97	1786	0.121533155	An Improved Hash <B>Code</B> <B>for</B> Scatter Storage Introduced is a hash coding method based on fixed-point division rather than multiplication or logical operations. This new method allows the hash table
2	98	3175	0.121533155	probabilities are updated in parallel <B>for</B> all <B>code</B> letters, using joint letter probabilities. Iterating the updating scheme results in improved estimates that finally lead to breaking the cipher. The
2	99	3064	0.12141173	Event Manipulation <B>for</B> Discrete Simulations Requiring Large Numbers of Events The event-manipulation system presented here consists of two major parts. The first part addresses the familiar
2	100	164	0.12103897	A Short Study of Notation <B>Efficiency</B> CACM August, 1960 Smith Jr., H. J. CA600802 JB March 20, 1978 9:02 PM

c) **Query 3:** parallel algorithms

Query id	Rank	Doc ID	Score	Text Snippet
3	1	2973	1.0088185	Sorting on a Mesh-Connected <B>Parallel</B> Computer Two <B>algorithms</B> are presented for sorting n^2 elements on an n X n mesh-connected processor array that require O(n) routing and comparison steps. The
3	2	3075	0.9137456	Fast <B>Parallel</B> Sorting <B>Algorithms</B> A <B>parallel</B> bucket-sort algorithm is presented that requires time O(log n) and the use of n processors. The algorithm makes use of a technique that requires more
3	3	2557	0.8322328	required to evaluate such products on ordinary serial computers as well as <B>parallel</B> computers is discussed. <B>Algorithms</B> are presented which properly parse such matrix sequences subject to the constraints
3	4	2266	0.8261164	zeros and their multiplicity are readily determined. At no point in the method is polynomial deflation used. CACM November, 1972 Patrick, M. L. <B>parallel</B> numerical <B>algorithms</B>, real polynomials, real
3	5	3156	0.8261164	, D. Graph theory, <B>parallel</B> processing, <B>algorithms</B>, transitive closure, connected component 5.25 5.32 6.22 CA790802 DB January 4, 1980 12:18 PM
3	6	1601	0.76765776	measurements are interpreted in a <B>parallel</B> processing environment. In such an environment the procedures obtained are superior to standard <B>algorithms</B>. CACM May, 1967 Shedler, G. S. CA670505 JB February
3	7	950	0.68843037	computers as can be expected to be available in the near future, much of numerical analysis will have to be recast in a more "<B>parallel</B>" form. By this is meant that serial <B>algorithms</B> ought to be replaced
3	8	1468	0.68843037	Syntax-Directed Interpretation of Classes of Pictures A descriptive scheme for classes of pictures based on labeling techniques using <B>parallel</B> processing <B>algorithms</B> was proposed by the author some

3	9	2433	0.594452	method and express <B>parallel</B> <B>algorithms</B> are described. CACM October, 1973 Millstein, R. E. array processing, parallelism detection, explicit parallelism, array allocation, <B>parallel</B> control
3	10	2570	0.594452	A Comparison of List Schedules for <B>Parallel</B> Processing Systems The problem of scheduling two or more processors to minimize the execution time of a program which consists of a set of partially
3	11	2289	0.5686563	implementations require from n^2 to n^3 steps. CACM September, 1972 Levitt, K. N. Kautz, W. H. graph theory, cellular logic-in-memory arrays, <B>parallel</B> processing, special purpose computers, <B>algorithms</B> for
3	12	1957	0.5663724	The List Set Generator: A Construct for Evaluating Set Expressions The list set generator is defined and <B>algorithms</B> for its use are given. The list set generator is a construct which may be added
3	13	2838	0.5339811	normally requires by performing garbage collection on a second processor in <B>parallel</B> with list processing operations, or on a single processor time-shared with them. <B>Algorithms</B> for recovering discarded
3	14	2692	0.48292387	Reentrant Polygon Clipping A new family of clipping <B>algorithms</B> is described. These <B>algorithms</B> are able to clip polygons against irregular convex plane-faced volumes in three dimensions, removing
3	15	141	0.4730892	Some Thoughts on <B>Parallel</B> Processing CACM October, 1960 Yarbrough, L. D. CA601007 JB March 20, 1978 8:16 PM
3	16	392	0.4730892	Comment on A Paper on <B>Parallel</B> Processing CACM February, 1961 Nekora, M. R. CA610206 JB March 17, 1978 12:58 AM
3	17	1302	0.4730892	<B>Parallel</B> Signaling Speeds for Data Transmission (Proposed American Stand ard) CACM March, 1965 CA650305 JB March 7, 1978 6:08 PM
3	18	2114	0.47197703	, and record retrieval are defined and from which some of the frequently used file structures such as inverted files, index-sequential files, and multilist files are derived. Two <B>algorithms</B> which
3	19	2723	0.47197703	Multiprocessing Compactifying Garbage Collection <B>Algorithms</B> for a multiprocessing compactifying garbage collector are presented and discussed. The simple case of two processors, one performing

3	20	2182	0.41395304	Interrupt Driven Programming CACM July, 1971 Zelkowitz, M. interrupts, supervisors, monitors, debugging, <B>parallel</B> processing, associative memories, microprogramming 3.51 4.32 4.42 CA710608 JB
3	21	2685	0.40970725	The <B>Parallel</B> Execution of DO Loops Methods are developed for the <B>parallel</B> execution of different iterations of a DO loop. Both asynchronous multiprocessor computers and array computers are
3	22	2714	0.39114887	Merging with <B>Parallel</B> Processors Consider two linearly ordered sets A, B, |A|=m, |B|=n, m<=n, and p, p<=m, <B>parallel</B> processors working synchronously. The paper presents an algorithm for merging A
3	23	2896	0.35849383	An Exercise in Proving <B>Parallel</B> Programs Correct A <B>parallel</B> program, Dijkstra's on-the-fly garbage collector, is proved correct using a proof method developed by Owicki. The fine degree of in
3	24	1262	0.35481688	Procedure-Oriented Language Statements to Facilitate <B>Parallel</B> Processing Two statements are suggested which allow a programmer writing in a procedure-oriented language to indicate sections of
3	25	1367	0.35481688	Character Structure and Character Parity Sense for <B>Parallel</B>-by-Bit Data Communication in ASCII* (Proposed American Standard) CACM September, 1966 CA660912 JB March 2, 1978 4:26 PM
3	26	1795	0.35481688	Optimal Code for Serial and <B>Parallel</B> Computation CACM December, 1969 Fateman, R. J. code optimization, sequencing of operations, detection of common subexpressions 4.12 CA691217 JB February 15, 1978 1:59 PM
3	27	1828	0.35481688	Synchronization in a <B>Parallel</B>-Accessed Data Base The following problem is considered: Given a data base which can be manipulated simultaneously by more than one process, what are the rules for
3	28	2700	0.35481688	Reduction: A Method of Proving Properties of <B>Parallel</B> Programs When proving that a <B>parallel</B> program has a given property it is often convenient to assume that a statement is indivisible, i.e. that
3	29	2007	0.3525926	<B>Algorithms</B> Policy/Revised August 1970 CACM August, 1970 CA700814 JB February 10, 1978 3:12 PM
3	30	371	0.34904885	Remarks on <B>Algorithms</B> 2 and 3, Algorithm 15 and <B>Algorithms</B> 25 and 26 CACM March, 1961 Wilkinson, J. H. CA610311 JB March 17, 1978 12:35 AM

3	31	1158	0.33452457	Program Structures for <B>Parallel</B> Processing Constructs for organizing and explicating <B>parallel</B> program segments are discussed as extensions to ALGOL 60. The constructs serve as meta-commands and
3	32	2785	0.30728042	language, and discusses some of the problems associated with <B>parallel</B> computer architectures. CACM March, 1975 Lawrie, D. H. Layman, T. Baer, D. Randal, J. M. GLYPNIR, Illiac IV, Programming
3	33	1796	0.29918474	Index by Subject to <B>Algorithms</B>, 1969 CACM December, 1969 This 1969 index is the first supplement to the Index by Subject to <B>Algorithms</B>, 1960 1968 (Comm. ACM 11, 12 (Dec. 1968), 827 830). CA691216 JB February 15, 1978 2:03 PM
3	34	2952	0.29568073	Functions Realizable with Word-<B>Parallel</B> Logical and Two's-Complement Addition Instructions CACM June, 1977 Warren, H. S. Jr. Boolean functions, two's-complement, sign propagation 4.0 6.32 CA
3	35	2895	0.292709	A Language for Formal Problem Specification A language for specifying the in tended behavior of communicating <B>parallel</B> processes is described. The specifications are constrain ts on the order in
3	36	1811	0.2897068	A Case Study in Programming for <B>Parallel</B>-Processors An affirmative partial answer is provided to the question of whether it is possible to program <B>parallel</B>-processor computing systems to
3	37	270	0.28207406	Techniques for Storage Allocation <B>Algorithms</B> CACM October, 1961 Kelley Jr., J. E. CA611011 JB March 16, 1978 12:50 PM
3	38	804	0.28207406	Exponentiation of Series (<B>Algorithms</B> 134) CACM July, 1963 Thacher Jr., H. C. CA630720 JB March 14, 1978 8:19 AM
3	39	1342	0.28207406	Transportation Problem (<B>Algorithms</B> 293 [H]) CACM December, 1966 Bayer, G. CA661207a JB March 2, 1978 2:30 PM
3	40	1660	0.28207406	Index By Subject To <B>algorithms</B>, 1960-1968 CACM December, 1968 CA681206 JB February 21, 1978 1:39 PM
3	41	1952	0.28207406	Index by Subject to <B>Algorithms</B>, 1970 CACM December, 1970 CA701211 JB February 9, 1978 2:50 PM
3	42	2325	0.28207406	Numerical Mathematics and Computer Science Numerical mathematics is viewed as the analysis of continuous <B>algorithms</B>. Four of the components of numerical mathematics are discussed. These are

3	43	2342	0.2644649	Interference Between Communicating <B>Parallel</B> Processes Various kinds of interference between communicating <B>parallel</B> processes have been examined by Dijkstra, Knuth, and others. Solutions have been
3	44	2851	0.2644649	Formal Verification of <B>Parallel</B> Programs Two formal models for <B>parallel</B> computation are presented: an abstract conceptual model and a <B>parallel</B>-program model. The former model does not distinguish
3	45	2401	0.25606704	On Shrinking Binary Picture Patterns A <B>parallel</B> processing algorithm for shrinking binary patterns to obtain single isolated elements, one for each pattern, is presented. This procedure may be
3	46	2865	0.25606704	Verifying Properties of <B>Parallel</B> Programs: An Axiomatic Approach An axiomatic method for proving a number of properties of <B>parallel</B> programs is presented. Hoare has given a set of axioms for
3	47	3059	0.25606704	Models for <B>Parallel</B> Processing WIthin Programs: Application to CPU:I/O and I/O:I/O Overlap Approximate queueing models for internal <B>parallel</B> processing by individual programs in a multiprogrammed
3	48	2080	0.2508934	requirements of program scheduling and resource allocation. The system nucleus simulates an environment in which program execution and input/output are handled uniformly as <B>parallel</B>, cooperating
3	49	3073	0.2508934	Communicating Sequential Processes This paper suggests that input and output are basic primitives of programming and that <B>parallel</B> composition of communicating sequential processes is a fundamental
3	50	1374	0.2468148	Evaluation of Determinant; Determinant Evaluation (<B>Algorithms</B> 41[F3]; 269[F3]) CACM September, 1966 Bergson, A. CA660909e JB March 23, 1978 4:29 PM
3	51	1851	0.2468148	Generator of Spanning Trees (<B>Algorithms</B> 354 [H]) CACM September, 1969 McIlroy, M. D. spanning trees, trees, graphs 5.32 CA690904 JB February 15, 1978 4:50 PM
3	52	1953	0.2468148	Exponential Integral Ei(x) (<B>Algorithms</B> 385 $S13)) CACM December, 1970 Redish, K. A. ANSI Fortran standard 4.0 4.22 CA701210 JB February 9, 1978 3:04 PM
3	53	2226	0.2468148	Further Evidence for the Analysis of <B>Algorithms</B> for the Zero-One Programming Problem The purpose of this note is to report computational experience additional to that recently summarized by Gue et

3	54	2417	0.2468148	Four Combinatorial <B>Algorithms</B> [G6] (Algorithm A466) CACM November, 1973 Ehrlich, G. permutations and combinations 5.39 CA731109 JB January 20, 1978 9:59 AM
3	55	2725	0.2468148	A Comparison of Simulation Event List <B>Algorithms</B> (Corrigendum) CACM August, 1975 Vaucher, J. C. Duval, P. CA750810 JB January 6, 1978 3:43 PM
3	56	2830	0.2468148	A Practitioner's Guide to Addressing <B>Algorithms</B> (Corrigendum) CACM September, 1976 Severance, D. G. Duhne, R. A. CA760909 JB January 4, 1978 8:43 AM
3	57	1551	0.2442833	On Compiling <B>Algorithms</B> for Arithmetic Expressions This paper deals with <B>algorithms</B> concerning arithmetic expressions used in a FORTRAN IV compiler for a HITAC-5020 computer having n accumulators
3	58	1569	0.2365446	statements <B>parallel</B> the structure and notation of the grammar. CACM July, 1967 Irwin, L. CA670704 JB February 28, 1978 9:01 AM
3	59	3044	0.2365446	different content. CACM November, 1978 Friedman, D. Wise, D. <B>Parallel</B> evaluation, suspending cons, Lisp, conditional forms, if-then-else, ambiguous function, infinite structures 4.2 4.13 4.32 5.24 CA
3	60	2884	0.23321806	Permutation Enumeration: Four New Permutation <B>Algorithms</B> Classical permutation enumeration <B>algorithms</B> encounter special cases requiring additional computation every nth permutation when generating
3	61	2902	0.21591799	Dynamic Memory Allocation in Computer Simulation This paper investigates the performance of 35 dynamic memory allocation <B>algorithms</B> when used to service simulation programs as represented by
3	62	2950	0.21591799	A Unifying Approach to Scheduling This paper presents a scheme for classifying scheduling <B>algorithms</B> based on an abstract model of a scheduling system which formalizes the notion of priority
3	63	3166	0.21591799	Computing Standard Deviations: Accuracy Four <B>algorithms</B> for the numerical computation of the standard deviation of (unweighted) sampled data are analyzed. Two of the <B>algorithms</B> are well-known in

3	64	1658	0.21155554	Analysis of <B>Algorithms</B> for the Zero-One Programming Problem This paper is concerned with a review and examination of several existing <B>algorithms</B> for the zero-one programming problem. Computational
3	65	2025	0.21155554	Student's t-Distribution; Jacobi Polynomials; Modified Romberg Quadrature; Factorial Analysis of Variance; (<B>Algorithms</B> 332,344,351,359) CACM July, 1970 Sale, A. H. J. Fortran standards
3	66	2222	0.21155554	Comment on London's Certification of Algorithm 245 CACM January, 1971 Redish, K. A. proof of <B>algorithms</B>, debugging, certification, metatheory, sorting, in-place sorting 4.42 4.49 5.24 5.31 CA
3	67	2362	0.21155554	Linear Equation Solver [F4] (Algorithm A423) CACM April, 1972 Moler, C. B. matrix <B>algorithms</B>, linear equations, Fortran, paged memory, virtual memory, array processing 4.22 4.32 5.14 CA720411 JB
3	68	2505	0.21155554	Reflection-Free Permutations, Rosary Permutations, and Adjacent Transposition <B>Algorithms</B> CACM May, 1973 Roy, M. K. permutation, permutation generation, scheduling, combinatorial analysis 5.39 CA
3	69	2863	0.21155554	VMIN-An Optimal Variable-Space Page Replacement Algorithm A criterion for comparing variable space page replacement <B>algorithms</B> is presented. An optimum page replacement algorithm, called VMIN, is
3	70	2942	0.21155554	An Algol-Based Implementation of SNOBOL 4 Patterns CACM July, 1977 Brownlee, J. N. patterns SNOBOL 4, pattern matching, string processing, pattern implementation, <B>algorithms</B> in Pascal 4.29 CA770710 JB December 28, 1977 8:15 AM
3	71	3061	0.21155554	Simulations of Dynamic Sequential Search <B>Algorithms</B> None CACM September, 1978 Tenenbaum, A. Searching, list processing, sequential searching, dynamic reordering, simulation 3.74 8.1 CA780911 DH January 29, 1979 6:30 PM
3	72	2740	0.20907786	semaphores and extended semaphores (queue semaphores). The number of <B>parallel</B> processes is carefully justified, and the various semaphore constructions are explained. The system is proved to be
3	73	1008	0.20697652	. The FASEB meeting is the largest scientific meeting held in the United States each year. The technique developed for FASEB can be applied to schedule any meeting with <B>parallel</B> sessions. CACM

3	74	1200	0.20697652	analogous to certain matrix operations, a <B>parallel</B> nomenclature is suggested for their classification. CACM September, 1965 Reily, E. D. Federighi, F. D. CA650906 JB March 6, 19787:33 PM
3	75	1380	0.20697652	of quasi-<B>parallel</B> processing. CACM September, 1966 Dahl, O. J. Nygaard, K. CA660907 JB March 2, 1978 6:21 PM
3	76	2727	0.20697652	Multiple Byte Processing with Full-Word Instructions A method is described which allows <B>parallel</B> processing of packed data items using only ordinary full-word computer instructions, even though the
3	77	1747	0.20485362	virtual computer are explained. Examples of applications of the criteria concern the reading of a time-of-day clock, the synchronization of <B>parallel</B> processes, protection in multiprogrammed systems
3	78	1846	0.20485362	On Simulating Networks of <B>Parallel</B> Processes in Which Simultaneous Events May Occur Some of the problems of simulating discrete event systems, particularly computer systems, on a conventional
3	79	2175	0.20485362	which the expression is to be executed, these subexpressions can be evaluated in serials, in <B>parallel</B>, or in a combination of these modes. This paper shows that expression execution time can be
3	80	2195	0.20485362	process is then discussed. The method described is suitable for <B>parallel</B> processing because the operations relative to each state can be computed in <B>parallel</B>, and the number of stages is equal to the
3	81	1873	0.19945648	Accelerating LP <B>Algorithms</B> It is shown how a novel method for computing (related) inner products can accelerate the pricing phase of LP <B>algorithms</B>. Other LP applications are indicated. CACM July
3	82	2272	0.19945648	Sorting by Natural Selection A family of sorting <B>algorithms</B> is proposed, the members of which make fuller use of the memory space and thus yield longer sorted strings. Extensive simulation results
3	83	1924	0.19710524	system. It is shown that carefully designed matrix <B>algorithms</B> can lead to enormous savings in the number of page faults occurring when only a small part of the total matrix can be in main memory at
3	84	2146	0.19710524	Optimizing the Polyphase Sort Various dispersion <B>algorithms</B> for the polyphase sorting procedure are examinedhe optimum algorithm based on minimizing the total number of unit strings read is

3	85	2273	0.19710524	Conversion of Decision Tables By Rule Mask Method Without Rule Mask Two <B>algorithms</B> for generating computer programs from decision tables are described. The <B>algorithms</B> allow handling limited entry
3	86	2283	0.19710524	Thinning <B>Algorithms</B> on Rectangular, Hexagonal, and Triangular Arrays In this report three thinning <B>algorithms</B> are developed: one each for use with rectangular, hexagonal, and triangular arrays
3	87	2903	0.19710524	Improving Programs by the Introduction of Recursion A new technique of program transformation, called "recursion in troduction," is described and applied to two <B>algorithms</B> which solve pattern
3	88	3006	0.19710524	Anomalies with Variable Partition Paging <B>Algorithms</B> Five types of anomalous behavior which may occur in paged virtual memory operating systems a redefined. One type of anomaly, for example
3	89	1392	0.18321247	Experience with FORMAC Algorithm Design Various facets of the design and implementation of mathematical expression manipulation <B>algorithms</B> are discussed. Concrete examples are provided by the
3	90	2490	0.18321247	Efficient <B>Algorithms</B> for Graph Manipulation [H] (Algorithm A447) Efficient <B>algorithms</B> are presented for partitioning a graph into connected components, biconnected components and simple paths. The
3	91	2679	0.18321247	Some Performance Tests of "quicksort" and Descendants Detailed performance evaluations are presented for six ACM <B>algorithms</B>: quicksort (No. 64), Shellsort (No. 201), stringsort (No. 207), "TREESORT
3	92	2997	0.18321247	<B>algorithms</B> use the "divide and conquer" technique and recursively apply a merge procedure for two nonin tersecting convex hulls. Since any convex hull algorithm requires at least O(n log n) operations
3	93	1341	0.17740844	correspondingly. The continued use of such a structure raises questions about its effects on the usefulness of future systems, particularly with regard to such trends as time sharing, <B>parallel</B>
3	94	1471	0.17740844	computer systems. These meta-instructions relate to <B>parallel</B> processing, protection of separate computations, program debugging, and the sharing among users of memory segments and other computing

3	95	1536	0.17740844	linkage to complex arithmetic subroutines. Evaluation of a function and derivative proceed in <B>parallel</B>, as in Wengert's procedure, but with the "imaginary" parts of variables declared complex
3	96	1554	0.17740844	<B>parallel</B> and perspective projections of four-dimensional hyperobjects rotating in four-dimensional space. The observed projections and their motions were a direct extension of three-dimensional
3	97	1603	0.17740844	/output timing, buffering, and task scheduling and provides <B>parallel</B> processing capability. User programs communicate with the monitor through a small set of meta-instruction which consists mostly of
3	98	1960	0.17740844	design rather than on implementation details. The main features of the system include the ability given to any user to schedule his own <B>parallel</B> processes using system primitive operations, the file
3	99	2376	0.17740844	nor will they run into a deadlock. CACM March, 1972 Habermann, A. N. <B>parallel</B> programming, multiprogramming, program correctness, process communication, process scheduling 4.30 4.32 4.39 4.9 CA
3	100	2514	0.17740844	procedure in a derivation or parse, using weighted programming matrices; he also has a choice of instance selection schemes (raster,random, <B>parallel</B>). Examples are given involving array languages

d. **Query 4:** parallel processor in information retrieval

Query id	Rank	Doc ID	Score	Text Snippet
4	1	2967	0.7082254	also addressed. CACM May, 1977 Stillman, N. J. Berra, P. B. associative memory, associative <B>processor</B>, content-addressable memory, graphics, <B>information</B> <B>retrieval</B>, data structures, software
4	2	634	0.5461165	Manipulation of Trees <B>in</B> <B>Information</B> <B>Retrieval</B>* CACM February, 1962 Salton, G. CA620209 JB March 20, 1978 3:44 PM
4	3	2973	0.5330611	. CACM April, 1977 Thompson, C. D. Kung, H. T. <B>parallel</B> computer, <B>parallel</B> sorting, <B>parallel</B> merge, routing and comparison steps, perfect shuffle. <B>processor</B> <B>in</B> terconnection pattern
4	4	1811	0.51177335	A Case Study <B>in</B> Programming for <B>Parallel</B>-Processors An affirmative partial answer is provided to the question of whether it is possible to program <B>parallel</B>-<B>processor</B> computing systems to
4	5	891	0.41518122	<B>retrieval</B> program as expensive and difficult (from a programming stand-point) to reconsider their position, for the present solution makes it possible to install an <B>information</B> <B>retrieval</B> program <B>in</B>
4	6	2530	0.40958735	An Algorithm for Extracting Phrases <B>in</B> a Space-Optimal Fashion [Z] (Algorithm A444) CACM March, 1973 Wagner, R. A. <B>information</B> <B>retrieval</B>, coding, text compression 3.70 5.6 CA730309 JB January
4	7	2965	0.40958735	An Optimal Evaluation of Boolean Expressions <B>in</B> an Online Query System CACM May, 1977 Hanani, M. Z. query, Boolean expression, <B>information</B> <B>retrieval</B>, file organization 3.5 3.70 3.74 CA770507 JB
4	8	2976	0.40958735	Approximating Block Accesses <B>in</B> Database Organizations CACM April, 1977 Yao, S. B. database, inverted file organization, database performance and measurement, <B>information</B> <B>retrieval</B>, query

4	9	3168	0.40958735	Comment on "An Optimal Evaluation of Boolean Expressions <B>in</B> an Online Query System." CACM October, 1979 Laird, P. Query, Boolean expression, <B>information</B> <B>retrieval</B>, file organization
4	10	2288	0.40007916	<B>information</B> <B>retrieval</B> system with a minimal search time and no redundant storage. Some important theorems on the consecutive <B>retrieval</B> property are proved <B>in</B> this paper. Conditions under which the
4	11	1927	0.38666227	<B>Information</B> Science <B>in</B> a Ph. Computer Science Program This report contains recommendations on a sample course curriculum <B>in</B> the general area of <B>information</B> organization and <B>information</B> system
4	12	2278	0.37855646	On Foster's <B>Information</B> Storage and <B>Retrieval</B> Using AVL Trees CACM September, 1972 Tan, K. C. binary trees, search trees, <B>information</B> storage, <B>information</B> <B>retrieval</B> 3.70 3.73 3.74 CA720912 JB January 27, 1978 4:10 PM
4	13	2838	0.35851765	normally requires by performing garbage collection on a second <B>processor</B> <B>in</B> <B>parallel</B> with list processing operations, or on a single <B>processor</B> time-shared with them. Algorithms for recovering discarded
4	14	2882	0.35653657	A Stochastic Evaluation Model for Database Organization <B>in</B> Data <B>Retrieval</B> Systems Experimental work <B>in</B> the valuation of large scale data <B>retrieval</B> systems has been scarce due to its difficulty and
4	15	1457	0.35578042	Data Manipulation and Programming Problems <B>in</B> Automatic <B>Information</B> <B>Retrieval</B> Automatic <B>information</B> <B>retrieval</B> programs require the manipulation of a variety of different data structures, including
4	16	2140	0.35020137	<B>Retrieval</B>-Update Speed Tradeoffs Using Combined Indices <B>In</B> a paper <B>in</B> the November 1970 Communications of the ACM, V. Y. Lum introduced a technique of file indexing named combined indices. This
4	17	1699	0.34822118	Experimental Evaluation of <B>Information</B> <B>Retrieval</B> Through a Teletypewriter Experiments designed to evaluate the capabilities of mechanized <B>information</B> <B>retrieval</B> systems, with emphasis on interactive

4	18	1601	0.34366482	iteration are computationally independent, making the methods of interest <B>in</B> a <B>parallel</B> processing environment. Convergence is insured by extracting the "best <B>information</B>" at each iteration
4	19	3169	0.34132278	Note on "An Optimal Evaluation of Boolean Expressions <B>in</B> an Online Query System." CACM October, 1979 Gudes, E. Hoffman, A. Query, Boolean expression, optimal evaluation, <B>information</B> <B>retrieval</B>
4	20	2516	0.3396688	Hierarchical Storage <B>in</B> <B>Information</B> <B>Retrieval</B> A probabilistic analysis is employed to determine the effect of hierarchical storage organizations on <B>information</B> <B>retrieval</B> operations. The data
4	21	1935	0.3378923	method described is binary <B>in</B> nature and offers new potential for <B>information</B> <B>retrieval</B> systems. CACM February, 1969 Arora, S. R. Dent, W. T. binary pattern, file examination, graph theory
4	22	2519	0.32636416	On the Problem of Communicating Complex <B>Information</B> The nature of the difficulty involved <B>in</B> communicating mathematical results between scientists using a computer based <B>information</B> <B>retrieval</B>
4	23	3012	0.32482263	The Use of an Interactive <B>Information</B> Storage and <B>Retrieval</B> System <B>in</B> Medical Research This paper presents the results of a study of the use of an interactive computerized storage and <B>retrieval</B>
4	24	1681	0.31271085	remote typewriter console. It has been developed for <B>retrieval</B> of documents from a computerized data base, the Moore School <B>Information</B> Systems Laboratory files. Requests are formulated <B>in</B> a
4	25	2114	0.3120005	A Formal System for <B>Information</B> <B>Retrieval</B> from Files A generalized file structure is provided by which the concepts of keyword, index, record, file, directory, file structure, directory decoding
4	26	2175	0.30691025	which the expression is to be executed, these subexpressions can be evaluated <B>in</B> serials, <B>in</B> <B>parallel</B>, or <B>in</B> a combination of these modes. This paper shows that expression execution time can be
4	27	1830	0.30682305	<B>Retrieval</B> Times for a Packed Direct Access Inverted File CACM October, 1969 Bayes, A. J. <B>information</B> <B>retrieval</B>, direct access memory, data base, inverted list 3.70 4.41 CA691016 JB February 15, 1978 1:27 PM

4	28	2865	0.29916105	for the use of auxiliary variables, which are added to a <B>parallel</B> program as an aid to proving it correct. The <B>information</B> <B>in</B> a partial correctness proof can be used to prove such properties as
4	29	1937	0.29438117	for specifying data <B>retrieval</B> and display requests. Data is displayed as tables and graphs produced <B>in</B> a format ready for publication. <B>In</B> this paper the statements of the request language and the
4	30	275	0.28660002	Dynamic Storage Allocation for an <B>Information</B> <B>Retrieval</B> System CACM October, 1961 Sams, B. H. CA611006 JB March 16, 1978 12:58 PM
4	31	651	0.28660002	A Survey of Languages and Systems for <B>Information</B> <B>Retrieval</B> CACM January, 1962 Grems, M. CA620108 JB March 20, 1978 4:36 PM
4	32	2070	0.28660002	A Formal System for <B>Information</B> <B>Retrieval</B> from Files CACM April, 1970 Hsiao, D. Harary, F. CA700414 JB February 13, 1978 2:37 PM
4	33	2990	0.28351814	Effective <B>Information</B> <B>Retrieval</B> Using Term Accuracy The performance of <B>information</B> <B>retrieval</B> systems can be evaluated <B>in</B> a number of different ways. Much of the published evaluation work is based
4	34	1675	0.27305824	first time are ineffectual, and that the factor is but a scale factor. CACM November, 1968 Korfhage, R. R. <B>information</B> <B>retrieval</B>, relevance, indexing, classification 3.70 3.71 3.74 CA681104 JB February 21, 1978 2:57 PM
4	35	2645	0.27148092	provide additional <B>information</B> about the program written <B>in</B> the first language and to output results estimating its efficiency. Processors for the two languages are also described. The first
4	36	2723	0.26939452	LISP-like list operations and the other performing garbage collection continuously, is thoroughly examined. The necessary capabilities of each <B>processor</B> are defined, as well as interprocessor
4	37	2497	0.26721078	branch would have resulted <B>in</B> an effective interrupt. CACM June, 1973 Hill, J. C. interrupts,supervisors, monitors, debugging, <B>parallel</B> processing, associative memories, microprogramming
4	38	2307	0.2662603	earlier file processing and of normal collection growth. The proposed procedures provide powerful tools for <B>information</B> <B>retrieval</B> and for the control of dynamic library collections <B>in</B> which new

4	39	1846	0.26329422	On Simulating Networks of <B>Parallel</B> Processes <B>in</B> Which Simultaneous Events May Occur Some of the problems of simulating discrete event systems, particularly computer systems, on a conventional
4	40	2455	0.26295358	A Generalization of AVL Trees A generalization of AVL trees is proposed <B>in</B> which imbalances up to (triangle shape) is a small integer. An experiment is performed to compare these trees with
4	41	1725	0.2572509	-addressed memories, ordered lists, ordered <B>information</B> <B>retrieval</B>, ordered <B>retrieval</B> theorem, column digit values, digit value variety, column sensing arrangement, digit value readout, digit variety
4	42	1032	0.25681505	Theoretical Considerations <B>in</B> <B>Information</B> <B>Retrieval</B> Systems <B>Information</B> storage and <B>retrieval</B> systems are composed of three major components: (a) identification of <B>information</B> and tagging it for
4	43	2631	0.25681505	An <B>Information</B>-Theoretic Approach to Text Searching <B>in</B> Direct Access Systems Using direct access computer files of bibliographic <B>information</B>, an attempt is made to overcome one of the problems
4	44	1959	0.2559755	programming language and supervisory system <B>in</B> which these concepts are implemented, is used to illustrated the new organization which is proposed for management <B>information</B> systems. CACM December
4	45	1359	0.25542647	Data Filtering Applied to <B>Information</B> Storage and <B>Retrieval</B> Applications Manipulation of data strings is the most complex processing function <B>in</B> <B>information</B> storage and <B>retrieval</B> applications
4	46	2947	0.2553496	system much more powerful than the sum of its parts. CACM July, 1977 Schneider, B. R. Jr. Watts, R. M. <B>information</B> <B>retrieval</B>, text editing, minicomputers, CRTs,time sharing, bibliographic search and
4	47	1236	0.2507608	several hundred different methods are available to analyze documents and search requests. This feature is used <B>in</B> the <B>retrieval</B> process by leaving the exact sequence of operations initially unspecified

4	48	2711	0.2507608	A Vector Space Model for Automatic Indexing <B>In</B> a document <B>retrieval</B>, or other pattern matching environment where stored entities (documents) are compared with each other or with incoming patterns
4	49	1747	0.24959369	virtual computer are explained. Examples of applications of the criteria concern the reading of a time-of-day clock, the synchronization of <B>parallel</B> processes, protection <B>in</B> multiprogrammed systems
4	50	1613	0.24941622	One-Pass Compilation of Arithmetic Expressions for a <B>Parallel</B> <B>Processor</B> Under the assumption that a <B>processor</B> may have a multiplicity of arithmetic units, a compiler for such a <B>processor</B> should
4	51	239	0.24565716	Inefficiency of the Use of Boolean Functions for <B>Information</B> <B>Retrieval</B> Systems CACM December, 1961 Verhoeff, J. Goffman, W. Belzer, J. CA611211 JB March 15, 1978 10:10 PM
4	52	292	0.24565716	An <B>Information</B> <B>Retrieval</B> Language for Legal Studies CACM September, 1961 Kehl, W. B. Horty, J. F. Bacon, C. R. T. Mitchell, D. S. CA610902 JB March 16, 1978 9:51 PM
4	53	1831	0.24565716	A Comment on Optimal Tree Structures CACM October, 1969 Stanfel, L. E. <B>information</B> <B>retrieval</B>, file searching, tree structures, double chaining 3.70 3.73 3.74 CA691015 JB February
4	54	2532	0.24565716	On Harrison's Substring Testing Technique CACM March, 1973 Bookstein, A. string, substring, hashing, <B>information</B> storage and <B>retrieval</B> 3.74 5.30 5.5 CA730307 JB January 24, 1978 10:30 AM
4	55	2561	0.24531764	A Heuristic Approach to Inductive Inference <B>in</B> Fact <B>Retrieval</B> Systems Heuristic procedures are presented which have been developed to perform inferences by generalizing from available <B>information</B>
4	56	2032	0.23892593	, tree structures, file structures, scatter tables, hashing functions, <B>information</B> <B>retrieval</B> 3.70 3.74 CA700705 JB February 13, 1978 8:49 AM
4	57	2493	0.23892593	. CACM June, 1973 Shneiderman, B. data base, reorganization, files, <B>information</B> <B>retrieval</B> 3.73 CA730607 JB January 23, 1978 2:14 PM
4	58	1960	0.23832633	Process Management and Resource Sharing <B>in</B> the Multiaccess System ESOPE The main design principles of the multiaccess system ESOPE are described. Emphasis is placed on basic ideas underlying the

4	59	2575	0.23623443	The Best-Match Problem <B>in</B> Document <B>Retrieval</B> CACM November, 1974 Van Rijsbergen, C. J. document <B>retrieval</B>, best match, clustering, file searching, matching, dissimilarity, hierarchy
4	60	1976	0.23291053	Multi-attribute <B>Retrieval</B> with Combined Indexes <B>In</B> this paper a file organization scheme designed to replace the use of the popular secondary index filing scheme (or inverted files on secondary key
4	61	2714	0.2318515	favorably with the previous best <B>parallel</B> merging algorithm, Batcher's algorithm, which requires n/p + ((m+n)/2p)log2 m steps <B>in</B> the general case and km/p + ((k+1)/2)(m/p)log2 m <B>in</B> the special case
4	62	2846	0.23022163	Compressed Tries This paper presents a new data structure, called a compressed trie or C-trie, to be used <B>in</B> <B>information</B> <B>retrieval</B> systems. It has the same underlying m-ary tree structure as a
4	63	1788	0.22707391	Toward a General <B>Processor</B> for Programming Languages Many efforts have been made to develop a better way of implementing a higher level programming language than by the construction of a whole new
4	64	3134	0.22261603	The Use of Normal Multiplication Tables for <B>Information</B> Storage and <B>Retrieval</B> This paper describes a method for the organization and <B>retrieval</B> of attribute based <B>information</B> systems, using the
4	65	2896	0.22205634	An Exercise <B>in</B> Proving <B>Parallel</B> Programs Correct A <B>parallel</B> program, Dijkstra's on-the-fly garbage collector, is proved correct using a proof method developed by Owicki. The fine degree of <B>in</B>
4	66	2160	0.22039233	Canonical Structure <B>in</B> Attribute Based File Organization A new file structure for attribute based <B>retrieval</B> is proposed <B>in</B> this paper. It allows queries involving arbitrary Boolean functions of
4	67	1367	0.21977878	Character Structure and Character Parity Sense for <B>Parallel</B>-by-Bit Data Communication <B>in</B> ASCII* (Proposed American Standard) CACM September, 1966 CA660912 JB March 2, 1978 4:26 PM
4	68	1711	0.21719776	keeping. <B>Information</B> is given on patents, copyrights and trade secret protection for programs, and the problem of using copyrighted material <B>in</B> <B>information</B> storage and <B>retrieval</B> systems, including the

4	69	2795	0.21719776	model. <B>In</B> generating sentences from meaning structures, the program employs both the <B>information</B> <B>retrieval</B> and deduction capabilities of the memory model. The model encompasses several diverse
4	70	1262	0.21610773	Procedure-Oriented Language Statements to Facilitate <B>Parallel</B> Processing Two statements are suggested which allow a programmer writing <B>in</B> a procedure-oriented language to indicate sections of
4	71	440	0.20856614	considerable discriminating power. Rules that can be applied generally to name <B>retrieval</B> systems have been developed <B>in</B> a methodological study of the linkage of vital and health records into family
4	72	1652	0.20711954	<B>in</B> area such as <B>information</B> <B>retrieval</B>, document classification, computer-aided teaching and text editing. This code, called IPC (<B>Information</B> Processing Code), is an 8-bit code set constructed so
4	73	1514	0.20635842	values of an adjustment statistic. An example evaluates the gain parameters for a typical <B>information</B> <B>retrieval</B> system. CACM November, 1967 Shumway, R. H. CA671106 JB February 26, 1978 2:59 PM
4	74	1828	0.20607825	Synchronization <B>in</B> a <B>Parallel</B>-Accessed Data Base The following problem is considered: Given a data base which can be manipulated simultaneously by more than one process, what are the rules for
4	75	2141	0.20479368	strings-sequence of like codes-by three methods and <B>in</B> four directions. Relationships are developed between compression alternatives to avoid comparing all of them. The technique has been used to
4	76	2543	0.20479368	Reducing the <B>Retrieval</B> Time of Scatter Storage Techniques A new method for entering and retrieving <B>information</B> <B>in</B> a hash table is described. The method is intended to be efficient if most entries
4	77	2412	0.2047143	Comment on Brent's Scatter Storage Algorithm CACM November, 1973 Feldman, J. A. Low, J. R. Hashing, <B>information</B> storage and <B>retrieval</B>, scatter storage, searching, symbol table
4	78	2552	0.2047143	A Note on When To Chain Overflow Items Within a Direct-Access Table CACM January, 1973 Bays, C. hash code, open hash, chaining, <B>information</B> <B>retrieval</B>, collision 3.7 4.9 CA730109 JB January
4	79	2781	0.2047143	<B>information</B> <B>retrieval</B> 3.7 CA750307 JB January 9, 1978 4:37 PM

4	80	1456	0.20312296	Storage and <B>Retrieval</B> of Aspects of Meaning <B>in</B> Directed Graph Structures An experimental system that uses LISP to make a conceptual dictionary is described. The dictionary associates with each
4	81	2623	0.20312296	other <B>information</B> coding techniques. CACM August, 1974 Hahn, B. file maintenance, <B>information</B> <B>retrieval</B>, utility programs, text compression, coding techniques, data storage, data management
4	82	2484	0.202765	results <B>in</B> different <B>information</B> structures, as list, tree, ring, etc. Thus the problem of <B>information</B> organization and storage is reduced to that of defining relations and formulating algorithms
4	83	3135	0.20197439	, chemical structure search, <B>information</B> <B>retrieval</B>, crystal -structure analysis, drug analysis and design 3.13 3.63 3.74 CA790401 DH May 21, 1979 10:50 AM
4	84	2451	0.20110689	Design of Tree Structures for Efficient Querying A standard <B>information</B> <B>retrieval</B> operation is to determine which records <B>in</B> a data collection satisfy a given query expressed <B>in</B> terms of data
4	85	2722	0.19743586	data structure for storage of <B>information</B> to be retrieved by associative searches. The k-d tree is defined and examples are given. It is shown to be quite <B>in</B> its storage requirements. A
4	86	1108	0.19457446	Digital Data <B>Processor</B> for Tracking the Partially Illuminated Moon* A study of lunar tracking techniques and fabrication of a breadboard to assess the feasibility of the best technique selected was
4	87	2377	0.19457446	A Hardware Architecture for Implementing Protection Rings Protection of computations and <B>information</B> is an important aspect of a computer utility. <B>In</B> a system which uses segmentation as a memory
4	88	406	0.19209304	The Use of Threaded Lists <B>in</B> Constructing a Combined ALGOL and Machine-Like Assembly <B>Processor</B> CACM January, 1961 Evans Jr., A. Perlis, A. J. Van Zoeren, H. CA610108 JB March 17, 1978 1:20 AM
4	89	1527	0.1893306	A Grammar Base Question Answering Procedure The subject of this paper is a procedure for the automatic <B>retrieval</B> of certain segments of stored <B>information</B>, either explicitly or implicitly

4	90	2991	0.18778704	, direct access method, randomizing, random access, file addressing, file organizations, file structures, scatter storage, search method, collisions, synonyms, clustering, <B>information</B> <B>retrieval</B>, open
4	91	1680	0.18740638	employing a display unit to interleave tutoring with other computer operations such as simulation, programming, and <B>information</B> <B>retrieval</B>. It is written <B>in</B> FORTRAN IV (G) for the IBM System
4	92	1194	0.18609634	, is reviewed briefly. Two principles are presented as paramount <B>in</B> the provision of <B>information</B> services: (1) easy accessibility to the <B>information</B> files by users unfamiliar with file organization
4	93	3075	0.18597947	Fast <B>Parallel</B> Sorting Algorithms A <B>parallel</B> bucket-sort algorithm is presented that requires time O(log n) and the use of n processors. The algorithm makes use of a technique that requires more
4	94	1742	0.18540506	is a key factor <B>in</B> display <B>processor</B> design. CACM June, 1968 Myer, T. H. display <B>processor</B> design, display system, computer graphics, graphic terminal, displays, graphics, display generator
4	95	1530	0.18354164	The ML/I Macro <B>Processor</B> A general purpose macro <B>processor</B> called ML/I is described. ML/I has been implemented on the PDP-7 and I. Atlas 2 computers and is intended as a tool to allow users to
4	96	2388	0.18319824	number of items <B>in</B> the tree. The binary trees grown by this algorithm sometimes have some branches longer than others; therefore, it is possible to reduce the average <B>retrieval</B> time by
4	97	2650	0.18319824	time proportional to |a|, the number of characters <B>in</B> a. The method should find applications <B>in</B> <B>information</B> <B>retrieval</B>, artificial intelligence, and spelling correction systems. CACM May
4	98	2746	0.18319824	. J. keywords and phrases, string pattern matching, bibliographic search, <B>information</B> <B>retrieval</B>, text-editing, finite state machines, computational complexity. 3.74 3.71 5.22 5.25 CA750607 JB
4	99	2895	0.18130824	A Language for Formal Problem Specification A language for specifying the <B>in</B> tended behavior of communicating <B>parallel</B> processes is described. The specifications are constrain ts on the order <B>in</B>

| 4 | 100 | 2266 | 0.18030451 | zeros and their multiplicity are readily determined. At no point <B>in</B> the method is polynomial deflation used. CACM November, 1972 Patrick, M. L. <B>parallel</B> numerical algorithms, real polynomials, real |