

# L A S T H O M E W O R K

## CS 667 : Homework 5(Due: Apr 25, 2013)

Problems 1-6 are for 200pts. You may replace some of them with Problem 7 or 8 for a total of 200.

**Problem 1.** (40 POINTS)

(a) We plan to sort  $n$  keys using  $p$  processors ala-PRAM like using merge-sort as follows. We split the  $n$  input keys into  $p$  subarrays each one of  $n/p$  keys (don't worry about ceilings or floors). Each processor deals with one of those subarrays. At the end we end up with  $p$  sorted sequences of  $n/p$  keys. One processor takes over to complete the sorting.

(i) If  $p = \sqrt{n}$ , fill-in the details by providing the parallel running-time of the approach  $T$  and the speedup  $s$  achieved over regular merge-sort. Explain and justify your answers.

(ii) If  $p = \lg n$ , repeat the questions of part (i).

(b) Instead of using merge-sort we use bubble-sort of the sorting of the  $p$  sequences. Repeat questions (i) and (ii) above.

**Problem 2.** (40 POINTS)

Let  $S = \langle x_1, x_2, \dots, x_n \rangle$  be a sequence of  $n$  distinct keys. The rank of  $x_1$  in the sequence  $S$  or  $r(x_1, S)$  is the number of keys less than  $x_1$  in  $S$ . The problem of sorting is equivalently the problem of determining the rank of each one of the  $n$  input keys.

Determine the rank of all keys in  $S$  in  $O(\lg n)$  time with a CRCW PRAM. How many processors did you use?

Sort the  $n$  keys in the same time with a CRCW PRAM. How many processors did you use?

Can you repeat the two questions above for an EREW PRAM? How would the answers change? Explain.

**Problem 3.** (40 POINTS)

(a) Give an EREW PRAM algorithm that merges two sorted arrays of size  $n/2$  with  $P = n/2$  in  $O(\lg n)$  time. You may assume that  $n$  is a power of two, and you may of course reuse prior or more recent results.

(b) Can you find the MAX of  $n$  keys with  $n^{7/6}$  processors in  $\Theta(1)$  time? Explain.

**Problem 4.** (40 POINTS)

We execute one query in Google and Bing. There are two pages of results for each. The query is CS 667 Algorithms. There a total of 20 results in two pages per search engine. Relevant documents are those that can positively identified from the available information (title, URL, context) as OUR COURSE. Everything else is NOT relevant.

(a) For each search engine, find and give the number of hits reported by each engine. Give the number of relevant document (read previous paragraphs) out of the 20 listed. Give the precision relative to the 20 documents reported for each engine. (These are the first 3 items in the table below.)

(b) Give 6-point effectiveness along the lines of page 33 of Handout 6 by generating a table similar to that of page 33. (This is item 4 in the table below. Items 5-8 can also be extracted.)

(c) Fill the table below. One point for a winner and 0 for the loser, 1 each for a tie. Who is the winner? Tie ?

	Values		Points	
	Google	Bing	Google	Bing
1. # Number of Hits reported (question (a))				
2. # Number of relevant docs among the 20				
3. Precision among the 20				
4. 6-point effectiveness				
5. 20% recall interpolated precision				
6. 60% recall interpolated precision				
7. 80% recall interpolated precision				
8. 3-point effectiveness(20,60,80)				
=====				
Number of point wins (sum)	-----	-----		

**Problem 5.** (20 POINTS)

**Kleinberg.** Find the hub/authority rank of the graph of Figure 1. Initial values will be  $1/N$  (not 1). Iterate as many times as needed for the error to be less than  $10^{-4}$ . (Do not forget scaling.)

**Problem 6.** (20 POINTS)

**PageRank.** Find the page rank of the graph of Figure 1. Initial values are  $1/N$ . Iterate as many times as needed for the error to be less than  $10^{-4}$ .

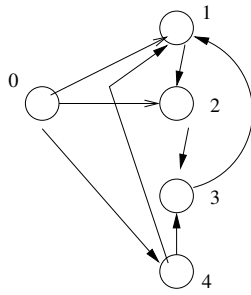


Figure 1: Problem 5-6 figure

**Problem 7.** (60 POINTS)

Use multithreading/multiprocessing, if you know how to do it, to implement the algorithm outlined in Problem 1 part (b). If you can't figure out how to implement bubble-sort, grab the code from my CS 435 web-page (section B4) and modify it as needed. I won't be able to help on multithreading. The data type used for testing would be randomly distributed doubles as in Homework 1. Time the running time of the sorting function implementation by capturing a running-time of ordinary bubble-sort, and then the modified one ON THE SAME INPUT sequence. A minimal interface needed is as follows.

```
% ./psort threads nkeys  
% java psort thread nkeys
```

**Problem 8.** (60 POINTS)

Implement the HITS and PageRank algorithms. The inputs will be graphs represented through an adjacency list. The command-line interface would be as follows.

```
% ./rank ranktype InitialValue Iterations InputFile
% java rank ranktype InitialValue Iterations InputFile
```

The command-line parameter `ranktype` takes one of two values: 0 if Kleinberg's HITS is used (with the scaling as otherwise shown on page XX of Subject YY) and 1 if the Brin and Page's PageRank algorithm is used (as shown on page ZZ of Subject YY). The second parameter `InitialValue` indicates how the initial values for the ranks will be computed. If it is 0 all ranks are initialized to 0, if it is 1 they are initialized to 1. If it is 2 they are initialized to  $1/N$ , where  $N$  is the number of web-pages (size of the graph.) If the value is a numeric integer value other than 0,1,2 then the ranks are initialized as `InitialValue` divided by 100. Thus an `InitialValue` equal to 50, initializes all ranks to  $50/100 = 0.5$ . Parameter `Iterations` runs the algorithms for that number of iterations. Parameter `InputFile` describes the input graph and it has the following form. The first line contains two numbers: the number of vertices (in the example below, this is equal to five) and the number of edges that follow on separate lines (i.e. six). In each line an edge  $(i, j)$  is presented by `i j`. The graph used in class in a lecture will be represented as follows. (Note that the graphs in class have vertices in the range  $1..n$ , whereas in this implementation, it is  $0..n - 1$ .)

```
4 4
0 2
0 3
1 0
2 1
```

Kleinberg might report, at the 14-th iteration, Authority/Hub pair values of

```
Base : 0 :A/H[ 0]=0.25000/0.25000 A/H[ 1]=0.25000/0.25000 A/H[ 2]=0.25000/0.25000 A/H[ 3]=0.25000/0.25000
Iterat : 1 :A/H[ 0]=0.50000/0.81650 A/H[ 1]=0.50000/0.40825 A/H[ 2]=0.50000/0.40825 A/H[ 3]=0.50000/0.00000
Iterat : 2 :A/H[ 0]=0.31623/0.94281 A/H[ 1]=0.31623/0.23570 A/H[ 2]=0.63246/0.23570 A/H[ 3]=0.63246/0.00000
Iterat : 3 :A/H[ 0]=0.17150/0.98473 A/H[ 1]=0.17150/0.12309 A/H[ 2]=0.68599/0.12309 A/H[ 3]=0.68599/0.00000
Iterat : 4 :A/H[ 0]=0.08771/0.99612 A/H[ 1]=0.08771/0.06226 A/H[ 2]=0.70165/0.06226 A/H[ 3]=0.70165/0.00000
Iterat : 5 :A/H[ 0]=0.04411/0.99902 A/H[ 1]=0.04411/0.03122 A/H[ 2]=0.70573/0.03122 A/H[ 3]=0.70573/0.00000
Iterat : 6 :A/H[ 0]=0.02209/0.99976 A/H[ 1]=0.02209/0.01562 A/H[ 2]=0.70676/0.01562 A/H[ 3]=0.70676/0.00000
Iterat : 7 :A/H[ 0]=0.01105/0.99994 A/H[ 1]=0.01105/0.00781 A/H[ 2]=0.70702/0.00781 A/H[ 3]=0.70702/0.00000
Iterat : 8 :A/H[ 0]=0.00552/0.99998 A/H[ 1]=0.00552/0.00391 A/H[ 2]=0.70709/0.00391 A/H[ 3]=0.70709/0.00000
Iterat : 9 :A/H[ 0]=0.00276/1.00000 A/H[ 1]=0.00276/0.00195 A/H[ 2]=0.70710/0.00195 A/H[ 3]=0.70710/0.00000
Iterat : 10 :A/H[ 0]=0.00138/1.00000 A/H[ 1]=0.00138/0.00098 A/H[ 2]=0.70711/0.00098 A/H[ 3]=0.70711/0.00000
Iterat : 11 :A/H[ 0]=0.00069/1.00000 A/H[ 1]=0.00069/0.00049 A/H[ 2]=0.70711/0.00049 A/H[ 3]=0.70711/0.00000
Iterat : 12 :A/H[ 0]=0.00035/1.00000 A/H[ 1]=0.00035/0.00024 A/H[ 2]=0.70711/0.00024 A/H[ 3]=0.70711/0.00000
Iterat : 13 :A/H[ 0]=0.00017/1.00000 A/H[ 1]=0.00017/0.00012 A/H[ 2]=0.70711/0.00012 A/H[ 3]=0.70711/0.00000
Iterat : 14 :A/H[ 0]=0.00009/1.00000 A/H[ 1]=0.00009/0.00006 A/H[ 2]=0.70711/0.00006 A/H[ 3]=0.70711/0.00000
```

and PageRank

```
Base : 0 :P[ 0]=0.25000 P[ 1]=0.25000 P[ 2]=0.25000 P[ 3]=0.25000
Iter : 1 :P[ 0]=0.25000 P[ 1]=0.25000 P[ 2]=0.14375 P[ 3]=0.14375
Iter : 2 :P[ 0]=0.25000 P[ 1]=0.15969 P[ 2]=0.14375 P[ 3]=0.14375
Iter : 3 :P[ 0]=0.17323 P[ 1]=0.15969 P[ 2]=0.14375 P[ 3]=0.14375
Iter : 4 :P[ 0]=0.17323 P[ 1]=0.15969 P[ 2]=0.11112 P[ 3]=0.11112
Iter : 5 :P[ 0]=0.17323 P[ 1]=0.13196 P[ 2]=0.11112 P[ 3]=0.11112
Iter : 6 :P[ 0]=0.14966 P[ 1]=0.13196 P[ 2]=0.11112 P[ 3]=0.11112
Iter : 7 :P[ 0]=0.14966 P[ 1]=0.13196 P[ 2]=0.10111 P[ 3]=0.10111
Iter : 8 :P[ 0]=0.14966 P[ 1]=0.12344 P[ 2]=0.10111 P[ 3]=0.10111
Iter : 9 :P[ 0]=0.14242 P[ 1]=0.12344 P[ 2]=0.10111 P[ 3]=0.10111
Iter : 10 :P[ 0]=0.14242 P[ 1]=0.12344 P[ 2]=0.09803 P[ 3]=0.09803
Iter : 11 :P[ 0]=0.14242 P[ 1]=0.12083 P[ 2]=0.09803 P[ 3]=0.09803
Iter : 12 :P[ 0]=0.14020 P[ 1]=0.12083 P[ 2]=0.09803 P[ 3]=0.09803
Iter : 13 :P[ 0]=0.14020 P[ 1]=0.12083 P[ 2]=0.09709 P[ 3]=0.09709
Iter : 14 :P[ 0]=0.14020 P[ 1]=0.12002 P[ 2]=0.09709 P[ 3]=0.09709
Iter : 15 :P[ 0]=0.13952 P[ 1]=0.12002 P[ 2]=0.09709 P[ 3]=0.09709
Iter : 16 :P[ 0]=0.13952 P[ 1]=0.12002 P[ 2]=0.09680 P[ 3]=0.09680
Iter : 17 :P[ 0]=0.13952 P[ 1]=0.11978 P[ 2]=0.09680 P[ 3]=0.09680
Iter : 18 :P[ 0]=0.13931 P[ 1]=0.11978 P[ 2]=0.09680 P[ 3]=0.09680
Iter : 19 :P[ 0]=0.13931 P[ 1]=0.11978 P[ 2]=0.09671 P[ 3]=0.09671
```



CS 667 Algorithms



Web Images Maps Shopping More - Search tools

About 8,330,000 results (0.28 seconds)

[CS 667 Design Techniques for Algorithms by A. V. Gerbessiotis ...](#)

[cs.njit.edu/alexg/courses/cs667/index.html](http://cs.njit.edu/alexg/courses/cs667/index.html)

Apr 4, 2013 - CS 667 Section 102 (Spring 2013). Course Information. Course E-mail: [alg667@cs.njit.edu](mailto:alg667@cs.njit.edu) [[alg667@oak.njit.edu](mailto:alg667@oak.njit.edu) DOES NOT WORK!] Time ...

[PDF CS 667 Frequently asked questions \(PDF\)](#)

[cs.njit.edu/alexg/courses/cs667/handouts/hand0.pdf](http://cs.njit.edu/alexg/courses/cs667/handouts/hand0.pdf)

File Format: PDF/Adobe Acrobat - [Quick View](#)

Jan 22, 2013 - CS 667: Frequently Asked Questions. 1. What's the complete name of the course? CS 667: Design techniques for algorithms. 2. Who takes this ...

[CS 667 - Computer Science Course Information](#)

[web.njit.edu](http://web.njit.edu) > [SPRING\\_2012](#) > List

Description, A. V. Gerbessiotis CS 667-101. Jan 5, 2012 Spring 2012. Course Information Handout 1. Sequential and parallel algorithms for numerical and ...

[PDF CS 667 Frequently asked questions \(PDF\)](#)

[cs.njit.edu/alexg/courses/cs667/OLD/S12/handouts/hand0.pdf](http://cs.njit.edu/alexg/courses/cs667/OLD/S12/handouts/hand0.pdf)

File Format: PDF/Adobe Acrobat - [Quick View](#)

Jan 5, 2012 - CS 667: Design techniques for algorithms. 2. Who takes this course? Students who have completed CS (or CIS) 610 at NJIT or have take a ...

[NJIT - New Jersey Institute of Technology CS 667 - Design Tech ...](#)

<https://www.myedu.com/Cs-667-Design-Techniques-Algorithms/.../s/2585993/>

CS 667 - Design Tech - Algorithms at New Jersey Institute of Technology is about Prerequisite: Cs 610. An Introduction To The Principles Of Major Design ...

[CS 667 Professors - MyEdu](#)

[www.myedu.com/Cs-667-Design-Tech-Algorithms/.../s/.../professor/](http://www.myedu.com/Cs-667-Design-Tech-Algorithms/.../s/.../professor/)

CS 667 professors, class schedule and professor ratings for Design Tech - Algorithms (CS 667) at New Jersey Institute of Technology (NJIT)

[NJIT CS 667 | Practice Exams, Lecture Notes, Textbooks, Study ...](#)

[www.coursehero.com](http://www.coursehero.com) > ... > NJIT > Computer Science (CS)

A. V. Gerbessiotis Sep 5, 2007 Course Information CS 667-101 Fall 2007 Handout 1 Sequential and parallel algorithms for numerical and combinatorial ...

[NJIT: CS 667: hand1](#)

[www.coursehero.com](http://www.coursehero.com) > New Jersey

V. A. Gerbessiotis Sep 5, 2007 Course Information CS 667-101 Fall 2007 Handout 1 Sequential and parallel algorithms for numerical and combinatorial ...

Results for similar searches

[Algorithm Recordings](#)

[www.algorithmrecordings.com/](http://www.algorithmrecordings.com/)

The Live It Up EP sees Sinister Souls flexing their hard, melodic drum & bass skills on 3 banging tracks. The EP wouldn't be complete without some signature ...


More results for [counter strike algorithm](#)

[MET CS 667 C1](#)

[people.bu.edu/kalathur/cs667\\_fall\\_04/metcs667.htm](http://people.bu.edu/kalathur/cs667_fall_04/metcs667.htm)

MET CS 667 C1 - Enterprise Java. (Main Campus, Wednesday, 6:00 - 9:00 PM). Instructor, Suresh Kalathur, Ph.D. Assistant Professor, Computer Science Dept. More results for [cs 667 algorithms](#)

Figure 2: Google for CS 667 Algorithms



■

[Go to Google Home](#)


[Web](#)
[Images](#)
[Maps](#)
[Shopping](#)
[More -](#)
[Search tools](#)


---

Page 6 of about 8,320,000 results (0.16 seconds)

[NJIT CS 667 Textbooks: Buy Sell Trade Design Tech - Algorithms ...](#)  
[www.locazu.com/NJIT-CS-667-textbooks...Algorithms/1888577](http://www.locazu.com/NJIT-CS-667-textbooks...Algorithms/1888577)  
 Buy, sell, trade used textbooks directly with other NJIT CS 667 students and save up to 90%.

[Spring 2012 Computer Science Schedule for Campus, Television ...](#)  
[www.cs.odu.edu/~ibf/spr12all.html](http://www.cs.odu.edu/~ibf/spr12all.html)  
 May 9, 2012 – 40, 28577, CS 101, Computers: An Introduction, 1300-1350, MWF, OCNPS 0200, GUPTA, R ..... Advanced Data Structures and Algorithms ...


 [Two Improved Range-Efficient Algorithms for F0 Estimation\\*](#)  
[www.cs.cityu.edu.hk/~okpoon/research/tamo07proc.pdf](http://www.cs.cityu.edu.hk/~okpoon/research/tamo07proc.pdf)  
 File Format: PDF/Adobe Acrobat - Quick View  
 by H Sun - Related articles  
 We present two new algorithms for range-efficient F0 estimating problem and ... So we seek for algorithms that run quickly using relatively small space. In particular, the time ..... Two Improved Range-Efficient Algorithms for F0 Estimation. 667 ...

 [From Logics to Algorithms - Department of Computer Science ...](#)  
[www.cs.rice.edu/~vardi/papers/wal07.pdf](http://www.cs.rice.edu/~vardi/papers/wal07.pdf)  
 File Format: PDF/Adobe Acrobat - Quick View  
 by MY Vardi - Cited by 23 - Related articles  
 vardi@cs.rice.edu ... emptiness-testing algorithms for these models of automata, this yields decision ... highly useful algorithm for LTL model checking [VW88a]. ..... Automata: From Logics to Algorithms. 667. T 0 u q0 v0 q0. T 1 u q0 v0 q0. T 2 u ...

[Scheme of Studies - PMAS-Arid Agriculture University Rawalpindi](#)  
[www.uaar.edu.pk/uit/scheme-of-studies.php?dept\\_id=31](http://www.uaar.edu.pk/uit/scheme-of-studies.php?dept_id=31)  
 60+ items – CS Core Courses (18/136), 6 courses. Supporting Courses ...  
 1 CS-323 Programming Fundamentals  
 3 CS-443 Data Structures and Algorithms

[CS URGE - What is Research in Computer Science?](#)  
[ora.org/ccc/csurge-what-is.php](http://ora.org/ccc/csurge-what-is.php)  
 Resources for Undergraduates - CS Graduate School ... undergraduate curriculum has an associated and active area of research, from algorithms, architecture, ...

[The Algorithm That Helps You Friend People You Don't Know](#)  
[ora.org/ccc/rh-friending.php](http://ora.org/ccc/rh-friending.php)  
 Computer scientists have developed an algorithm that uses the structure of a social ... Phone: 202-234-2111 | Fax: 202-667-1066 | E-mail: webmaster @ ora.org.

 [The Efficiency of Algorithms – Misconceptions Judith ... - Cite...](#)  
[citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.136...](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.136...)  
 File Format: PDF/Adobe Acrobat - Quick View  
 by J Gal-Ezer - Cited by 3 - Related articles  
 includes all the basic elements of traditional CS programs. At its core ... efficiency of algorithms for the first time in the framework of a high school curriculum, it was crucial for its ..... International Journal of Science Education, 18, 6, 653-667, 21.

[CS 667 Design Techniques for Algorithms by A. V. Gerbessiotis ...](#)  
[cs.njit.edu/alexg/courses/cs667/OLD/S12/index.html](http://cs.njit.edu/alexg/courses/cs667/OLD/S12/index.html)  
 Apr 18, 2012 – CS 667 Section 101 (Spring 2012). Course Information. Course E-mail: alg667@cs.njit.edu [alg667@oak.njit.edu DOES NOT WORK!] ...


 [Incremental Constructions con BRIO - Computer Science Division](#)  
[www.cs.berkeley.edu/~jrs/meshpapers/BRIO.pdf](http://www.cs.berkeley.edu/~jrs/meshpapers/BRIO.pdf)  
 File Format: PDF/Adobe Acrobat - Quick View  
 by N Amenta - 2003 - Cited by 80 - Related articles

Figure 3: Google for CS 667 Algorithms





WEB IMAGES VIDEOS MAPS NEWS MORE

CS 667 Algorithms



29,900 RESULTS Any time ▾

### [CS 667: Introduction to Parallel Algorithms](https://www.student.cs.uwaterloo.ca/~cs667)

<https://www.student.cs.uwaterloo.ca/~cs667>

**CS 667: Introduction to Parallel Algorithms** . This course will no longer be offered in Winter 1999; it will be replaced by CS 760L, also to be taught by Naomi ...

### [Design Techniques for Algorithms - Computer Science Course ...](http://web.njit.edu/cs/cs_courses/index.php?cno=234&s=SPRING_2012)

[web.njit.edu/cs/cs\\_courses/index.php?cno=234&s=SPRING\\_2012](http://web.njit.edu/cs/cs_courses/index.php?cno=234&s=SPRING_2012) ▾

Course No. **CS 667**: Sections: 102: Title: Design Techniques for **Algorithms**: Course Website: Prerequisite(s)-Instructor: Alexandros Gerbessiotis; Office Room No. : GITC ...

### [CS 667 Design Techniques for Algorithms by A. V. Gerbessiotis ...](http://cs.njit.edu/alexg/courses/cs667)

[cs.njit.edu/alexg/courses/cs667](http://cs.njit.edu/alexg/courses/cs667)

A. Announcements . 4/4 HW4 emails have been acknowledged (as of 17:18, Thu 4/4); 4/4 Office Hours today April 4, 2013 I might be late; please click the link at the top ...

### [CS 667 Design Techniques for Algorithms by A. V. Gerbessiotis ...](http://cs.njit.edu/alexg/courses/cs667/OLD/S12)

[cs.njit.edu/alexg/courses/cs667/OLD/S12](http://cs.njit.edu/alexg/courses/cs667/OLD/S12)

A. Announcements . Apr19: HW5 All HW5 (Problem 8 and 9) submissions have been graded. Emails have been sent out around 2pm on Apr 19. Solutions for P1-7 will be ...

### [CS667-12 Optimization Algorithms - DSPCSP Pages](http://www.dspscsp.com/poly/lect12.htm)

[www.dspscsp.com/poly/lect12.htm](http://www.dspscsp.com/poly/lect12.htm) ▾

**CS 667 Optimization Algorithms**. What are evolutionary algorithms (EA)? NOT neural networks; Use long time biological intelligence; Key ideas from biological evolution;

### [CS667-05 Soft Neurons and the LMS Algorithm - DSPCSP Pages](http://www.dspscsp.com/poly/lect05.htm)

[www.dspscsp.com/poly/lect05.htm](http://www.dspscsp.com/poly/lect05.htm) ▾

**CS 667 Soft Neurons and the LMS Algorithm**. Widrow's adaptive filter; Noise cancellation problem; FIR filters; Adaptive FIR filters; Minimization of energy; ADALINE;

### [Consequences and Limits of Nonlocal Strategies](https://cs.uwaterloo.ca/~cleve/courses/F09CS667/Lec7toLec9Qip08.ppt)

<https://cs.uwaterloo.ca/~cleve/courses/F09CS667/Lec7toLec9Qip08.ppt>

Introduction to Quantum Information Processing CS 467 / **CS 667 Phys 667** / Phys 767 C&O 481 / C&O 681 Lecture 7 (2008) Richard Cleve DC 2117 [cleve@cs.uwaterloo.ca](mailto:cleve@cs.uwaterloo.ca)

### [A.V. Gerbessiotis: Courses \(CS list\)](http://web.njit.edu/~alexg/courses/index.html)

[web.njit.edu/~alexg/courses/index.html](http://web.njit.edu/~alexg/courses/index.html)

Graduate level courses. **CS 667**: Design Techniques for **Algorithms**. Fall 2006, Fall 2007, Fall 2009, Spring 2012, Spring 2013. **CS 610**: Data Structures and **Algorithms**.

### [Cornell : CS 667 : moore88](http://www.coursehero.com/file/1869957/moore88)

[www.coursehero.com/file/1869957/moore88](http://www.coursehero.com/file/1869957/moore88)

Register now to access 7 million high quality study materials (What's Course Hero?) Course Hero is the premier provider of high quality online educational ...

### [NJIT - Graduate Catalog: Computer Science](http://catalog.njit.edu/graduate/programs/computerscience.php)

[catalog.njit.edu/graduate/programs/computerscience.php](http://catalog.njit.edu/graduate/programs/computerscience.php) ▾

**CS 667**: Design Techniques for **Algorithms** (3 credits) ...

Figure 4: Bing for CS 667 Algorithms



### [CSE 667 - Stony Brook University - Department of Computer Science](#)

[www.cs.sunysb.edu/graduate/courses/cse667.html](http://www.cs.sunysb.edu/graduate/courses/cse667.html) ↕

Why CS @ SBU ; Our Policies; ... CSE 667 : Back to Special Topics Courses: Course:

CSE667 : Title: **Algorithms**: Credit Information: 2 credits: M.S. Program ;

### [Course Requirements | Rutgers Business School](#)

[business.rutgers.edu/mit/old-course-requirements](http://business.rutgers.edu/mit/old-course-requirements) ↕

CS 667 Design Tech-**Algorithms**; CS 670 Artificial Intelligence; CS 680 Linux Kernel Programming; CS 684 Software Test & Quality Assurance; CS 698 ST: Software ...

### [QIC 700s](#)

[www.ucaledar.uwaterloo.ca/SA/GRAD/0910/GRDcourse-QIC.html](http://www.ucaledar.uwaterloo.ca/SA/GRAD/0910/GRDcourse-QIC.html) ↕

(Cross-listed with CO 681, PHYS 767, AMATH 871, CS 667) Review of basics of quantum information and computational complexity; ... QIC 823 Quantum **Algorithms** (0.50) LEC:

### [CS 665 - Department of Computer Science, Cornell University](#)

[www.cs.cornell.edu/Courses/cs665](http://www.cs.cornell.edu/Courses/cs665) ↕

be on practical rendering **algorithms** for real applications. ... Any of CS 465/466, 467/468, or 667 are acceptable pre-requisites. If you have not taken these

### [Course Work](#)

[www.cs.iastate.edu/~svskati/courses.html](http://www.cs.iastate.edu/~svskati/courses.html) ↕

CS 667 Design Techniques for **Algorithms** CS 704 Sequencing and Scheduling CS 811 Computability and Complexity CS 744 Data Mining and Management in Bio- ...

### [Project Abstracts for cs 667, co 681, ph 767, am 871 \(Fall 2009\)](#)

[en.convdocs.org/docs/index-32937.html](http://en.convdocs.org/docs/index-32937.html)

Project Abstracts for CS 667, CO 681, PH 767, AM 871 (Fall 2009) This file will be updated as more abstracts are received. If I didn't receive a recent email about ...

### [Dijkstra Algorithm implementation in Java](#)

[cs.nyu.edu/~vs667/development/~DijkstraAlgorithm](http://cs.nyu.edu/~vs667/development/~DijkstraAlgorithm) ↕

Program Summary: This is my implementation of the Dijkstra **Algorithm**. Dijkstra's **algorithm**, conceived by Dutch computer scientist Edsger Dijkstra in 1959,[1] is a ...

### [NJIT - Courses: Computer Science](#)

[catalog.njit.edu/courses/cs.php](http://catalog.njit.edu/courses/cs.php) ↕

CS 667 - Design Techniques for **Algorithms** (3 credits) Prerequisite: CS 610. An introduction to the principles of major design techniques in **algorithms**.

### [Graduate Course Descriptions - NDSU Computer Science](#)

[cs.ndsu.edu/gradcourses.htm](http://cs.ndsu.edu/gradcourses.htm) ↕

Basic principles and **algorithms** of dynamic programming as applied to sequential decision problems in CS and OR. Prereq: Math 166. 667: **Algorithm** Analysis: 3; ...

### [My Home Page](#)

[www.cse.unr.edu/~tippabha/index.html](http://www.cse.unr.edu/~tippabha/index.html)

Graduate Student in **Computer Science** at UNR. ... CS 790K: Genetic **Algorithms**: 3.00: Sushil J. Louis: Fall '97: 9: ... CS 667: Theory of Computation: 3.00:

Figure 5: Bing for CS 667 Algorithms