L1 4800

Algorithms

abhi shelat
Sep 9 2016
anthem
let me intro myself
first goal: create an amazing learning experience
second goal: instill enthusiasm for this area
third goal: help prepare you for a job in cs
caveat emptor
This was one of the most brutally difficult courses I have taken. Almost every homework ended with me staying up all night before it was due in order to get it finished. However, all told, this has also been one of the most worthwhile classes I have taken. The work is very difficult, but because of that it was even more rewarding every time I solved a problem. Abhi is incredibly enthusiastic about the topic and really does his best to get the class to actually learn something. He also really knows the subject, and is almost always able to quickly and accurately respond to any student questions.
Algorithms has **single-handedly been the most difficult, yet most rewarding class** I have ever taken. Ever. The class was taught in the best way for me to learn. Personally, I am able to work my best when I can work on projects on my own schedule. Thus, having all of the work in a 'pset' format was optimal. The homeworks were hard, but they're completely worth the effort; likewise for the exams. Prof. Shelat is really enthusiastic about Algorithms, and that really came through in his teaching. The lectures were interesting and very informative; it was very helpful to have the annotated PDF's and screencasts. Also, I appreciated all of the ways that Shelat made himself available to the students. Piazza was a great way to answer questions, and Shelat was very quick to respond to emails. Despite the high difficulty of the course, there was never a moment where I felt that I couldn't solve a problem. By that, I don't mean that the problems were easy or simple, but rather, I knew that if I put in enough time and effort, I would eventually be able to solve them. This was a significant change in my learning paradigm because Algorithms is the first class I've taken where I've wanted to solve the problems we were given distinctly for the acting of solving them and knowing how they work. There was no busy-work in this class; all of it was meaningful. Taking algorithms this semester was the reason why I was able to pass the programming interviews I had this fall. I supremely enjoyed this class and I am very glad that I took it.
Shelat turned this formerly-easy class into pure hell. All the assignments have been stupid hard, throw-up-your-hands-in-frustration level difficulty. And they rarely have anything to do with the lectures. And the problems are poorly written. And the assignment grading is excessively harsh, frequently arbitrary, and often inconsistent. And Shelat has been completely unresponsive to the many student complaints about all this. This has been the worst kind of hard class; the kind where you work insanely hard only to accomplish nothing meaningful. Bottom line: Shelat should never be allowed to teach an undergraduate course ever again, at any school!
50% hw

50% exams
what is this course about?
Theme

Small problems are easy to solve
Theme

Small problems are easy to solve

Solve big problems by making them into smaller ones
to convince through reason is a good mark of understanding
great pyramid at giza 2500bc

image from wikimedia
“how much granite/glass do i need?”
algorithm to compute $\pi$
red perimeter $< \pi d$
red perimeter $< \pi d <$ blue perimeter
$\sqrt{3}$
\( x^2 - 3 = 0 \)
\frac{265}{153} \approx \sqrt{3}
red perimeter $< \pi d <$ blue perimeter
Using 96-gon, Archimedes

\[3 \frac{10}{70} > \pi > 3 \frac{10}{71}\]
how to analyze this approach?
\[ \pi = \frac{9801}{\sqrt{8}} \left( \sum_{n=0}^{\infty} \frac{(4n)!}{(n!)^4 (1103 + 26390n)} \right)^{-1} \]
\[ \pi = \frac{9801}{\sqrt{8}} \left( \sum_{n=0}^{\infty} \frac{(4n)!(1103 + 26390n)}{(n!)^4 396^{4n}} \right)^{-1} \]

\[ n=0 \]
\[ \pi = \frac{9801}{\sqrt{8}} \left( \sum_{n=0}^{\infty} \frac{(4n)! (1103 + 26390n)}{(n!)^4 396^{4n}} \right)^{-1} \]

\[ n = 0 \]

\[ \pi \approx_0 \frac{9801}{\sqrt{8}} [1103]^{-1} \]

3.14159273001330576017
\[ \pi = \frac{9801}{\sqrt{8}} \left( \sum_{n=0}^{\infty} \frac{(4n)!(1103 + 26390n)}{(n!)^4 396^{4n}} \right)^{-1} \]

\[ n = 1 \]
\[ \pi = \frac{9801}{\sqrt{8}} \left( \sum_{n=0}^{\infty} \frac{(4n)! (1103 + 26390n)}{(n!)^4 396^{4n}} \right)^{-1} \]

\[ \pi \approx_1 \frac{9801}{\sqrt{8}} \left[ 1103 + \frac{24 \cdot 27493}{396^4} \right]^{-1} \]

\[ 3.14159265358979387799890582630 \]
benefits?
good algorithms touch every aspect of our lives
good algorithms defend freedom
what skills do you need for this course?
precision
creativity
ingenuity
how to learn in this class
no cookbook
develop general problem solving skills
understand known techniques
work with your peers
work with your peers

but do not copy
You searched for algorithms. 118 matches

**Algorithms, Part I**
Princeton University

**Approximation Algorithms Part I**
École normale supérieure
LATEX
The Not So Short Introduction to \LaTeXe

Or \TeXe\ in 157 minutes

by Tobias Oetiker
%--------change this by adding your uva id into the {}
\def\yourname{}

% no need to change anything in this section ----------------------------------
\def\homework(1) % 0 for solution, 1 for problem-set only
\def\due(\fri jan 29, 2016 at 5p)
\def\due(location) {via \url{https://church.cs.virginia.edu/~shelat/16s-4102} [submission site]}
\def\number(0)
\def\prof(abhi shelat)
\def\course{\url{https://www.cs.virginia.edu/~shelat/16s-4102} [cs4102 - algorithms - s'16]}
\documentclass[11pt]{article}

% == no need to edit any of this stuff
%

% standard installations of latex include all of the files that are referenced in this section. However, % if you are having compile problems, consider commenting some of these commands out
\usepackage[colorlinks, urlcolor=blue]{hyperref}
\usepackage[osf]{mathpazo}
\usepackage{amsmath,amsfonts,graphicx}
\usepackage{latexsym}
\usepackage[geometry]{top=1in, bottom=1.4in, left=1.5in, right=1.5in, centering}
\usepackage{color}
\definecolor{mdb}{rgb}{0.3,0,0.02}
\definecolor{cit}{rgb}{0.05,0,0.45}
\makeatletter
\newcommand\yourname{\@nameuse{\@tempcnta\let\@tempcnta=0\verbatim@process}}
\makeatother

% == should be no need to edit anything in this section ======================
\newenvironment{proof}{\par
\noindent \hspace{1em}($\$\Box\$\$)\hspace{1em}}{\proof@h{1em}}
\newcommand\qed{\$\Box$}
\newcommand\alg{\alg algorithm}
\newcommand\handout{
\newcommand{\page}{\makeatletter\iftex{}{\@pfn\page}}
\newcommand{\Hnumber\makeatletter\iftex{}{\@pfn{\@arabic{\value{page}}}}\makeatother
\begin{center}%
\vbox{\setlength{\columnwidth}{\textwidth-2\columnsep}
% columnwidth \{sc\} course \{abhi shelat\} \vfill %
% \columnwidth \{sc due \MakeLowercase{\due(\fri jan 29, 2016 at 5p) \due(location)\vfill \Hnumber\{huge\color{mdb}\} \number(1) \prof}}
\end{center}
\end{center}}
About 189,000 results (0.37 seconds)

TeXShop
pages.uoregon.edu/koch/texshop/ ▼ University of Oregon ▼
TeXShop (v 3.59) Release 01/01/2016. (Mountain Lion or Higher Strongly Recommended). (for Lion, Mountain Lion, Mavericks, Yosemite, El Capitan) ...

Obtaining TeXShop
Obtaining TeXShop. If you just want to upgrade to the latest ...

Installing
Direct Download: TeXShop 3 for Lion | Lion Source | TeXShop 2 ...

More results from uoregon.edu »
Introduction
Page headers and footers
What is VerTex(Fancyhdr)
Simple use of VerTex(Fancyhdr)
A simple example
An example of two-sided printing
Redefining VerTex(plain) style
The default layout
The scoop on LaTeX's marks
Dictionary style headers
Fancy layouts
Two book examples
Special page layout for float pages
Those blank pages
VerTex(N) of VerTex(M) style page numbers
Chapter or section related page numbers
When to change the headers and footers?
Headers and footers induced by the text
Package for extra marks in LaTeX
A movie
Thumb-indexes
Float placement
Multipage Floats
Contact information

\extramarks{}{(Continued on next page)\ldots}
Some text that may or may not...
\extramarks{Continued}
\end{verbatim}

\Cmd{extramarks}
\Cmd{firstmark}
\Cmd{lastmark}
\Cmd{firstleftmark}
\Cmd{lastrightmark}
\Cmd{firstmark} gives you the first \$m_1$ value and \Cmd{lastmark} gives you the last \$m_2$ value of the current page.
\Cmd{firstleftmark} and \Cmd{lastrightmark} give you the \Cmd{firstmark} and \Cmd{lastmark} values for the current page.

\begin{verbatim}
1-ni-tion would
\end{verbatim}

\begin{verbatim}
Underfull \hbox (badness 5077) in paragraph at lines 1088--1095
\end{verbatim}

\begin{verbatim}
/cm10/be /cm10/arabic(page) /cm10/but you can give this def-i-
ni-tion your-self after the
[16] [17] [18] [19] [20] [21] [22] [23] [24]
No file fancyhdr.ind.
[25] (fancyhdr.aux)
\LaTeX Warning: Label(s) may have changed. Rerun to get cross-references right.
\end{verbatim}
Submitting HW

gradescope
Honor Policy

I, ______, do hereby certify on my honor that during this course,

1. I shall write my answers entirely by myself, and neither share nor request text, code, or drawings.

2. I will not give or derive assistance from any unauthorized sources or the web.
counting
1 stand
1. stand

2. set your “number” to one
1. stand
2. set your “number” to one
3. greet a neighbor (pause if odd person out)
1. stand
2. set your “number” to one
3. greet a neighbor (pause if odd person out)
4. if you are older, give “number” and sit
   if you are younger, add “numbers”
1. stand

2. set your “number” to one

3. greet a neighbor (pause if odd person out)

4. if you are older, give “number” and sit
   if you are younger, add “numbers”

5. if you are standing & you have a neighbor, go to 3
1 stand 2 set 3 greet 4 sit/add 5 repeat

lets analyze this alg
how fast does it work:

1 stand
2 set
3 greet
4 sit/add
5 repeat
how fast does it work:

\[ T(n) \]  # steps to finish in a room with n people
1. stand
2. set
3. greet
4. sit/add
5. repeat
how fast does it work:

$$T(n) = 1 + 1 + T(\left\lfloor n/2 \right\rfloor)$$
what is a recurrence?
what is a recurrence?

\[ T(n) = T(\lceil n/2 \rceil) + 2 \]

\[ T(1) = 3 \]
solve a simpler case when \( n \) is a power of 2.

\[
T(2^k) = 2 + T(2^k - 1)
\]
\[ T(2^k) = 2 + T(2^{k-1}) \]

“intuition here”
\[ T(2^k) = 2 + T(2^{k-1}) \]
\[ = 2 + 2 + T(2^{k-2}) \]
\begin{equation}
T(2^k) = 2 + T(2^{k-1})
\end{equation}

\begin{equation}
= 2 + 2 + T(2^{k-2})
\end{equation}

\begin{equation}
\underbrace{2 + 2 + \cdots + 2}_{k}
+ T(2^0)
\end{equation}

“intuition here”
\[
T(2^k) = 2 + T(2^{k-1})
\]
\[
= 2 + 2 + T(2^{k-2})
\]
\[
= 2 + 2 + \cdots + 2 + T(2^0)
\]
\[
= 2k + T(1)
\]