

# EECS 440 System Design of a Search Engine

Winter 2021

Lecture 2: Search engine basics

Nicole Hamilton

[https://web.eecs.umich.edu/~nham/  
nham@umich.edu](https://web.eecs.umich.edu/~nham/nham@umich.edu)

# Agenda

1. Course details.
2. HW1 MostPositiveSubsequence( ).
3. History of the web.
4. Search basics.
5. Your project.

# Agenda

1. Course details.
2. HW1 MostPositiveSubsequence( ).
3. History of the web.
4. Search basics.
5. Your project.

# details

1. HW2 is posted, asks that you create a Gantt chart for a plan to apply to graduate school, due Sunday.
2. Please download and install GanttProject <https://www.ganttproject.biz/> in preparation for next lecture on project planning.
4. Who needs a team? Is the speed dating helpful?
5. Questions on group photos?

# Agenda

1. Course details.
2. HW1 MostPositiveSubsequence( ).
3. History of the web.
4. Search basics.
5. Your project.

# Homework

Write a C++ function that can scan a sequence of  $N$  integers in an array  $A$ , returning the sum and the left and right indices of the most positive subsequence.

For example, for the sequence

$\{-1, 3, 5, 6, -2, -4, 1, 7, -15, 12, 7, -5\}$

the best sum = 20, left = 1, right = 10.

# Homework

Was it ambiguous?

Anything not specified?

# Homework

Was it ambiguous? Yes.

Anything not specified?

1. What happens if a null sequence is given?
2. What if all the numbers are negative or zero?
3. Does it have to be the shortest sequence?
4. What if two sequences are identical?
5. Does performance matter?



# Early iterations

In earlier iterations there was even more ambiguity. I only gave a few examples of what the program should print.

Some students did everything in main.

If they broke it out as a separate procedure, some passed pointers to the bestL and bestR, some passed by reference.

Some created structs or classes to hold results.

Some printed in the procedure, some in main.

Some broke things out into separate .cpp and .h files, some did not.

# What you saw

You saw a less ambiguous problem, to make it suitable for autograding.

Rules you were asked to discover:

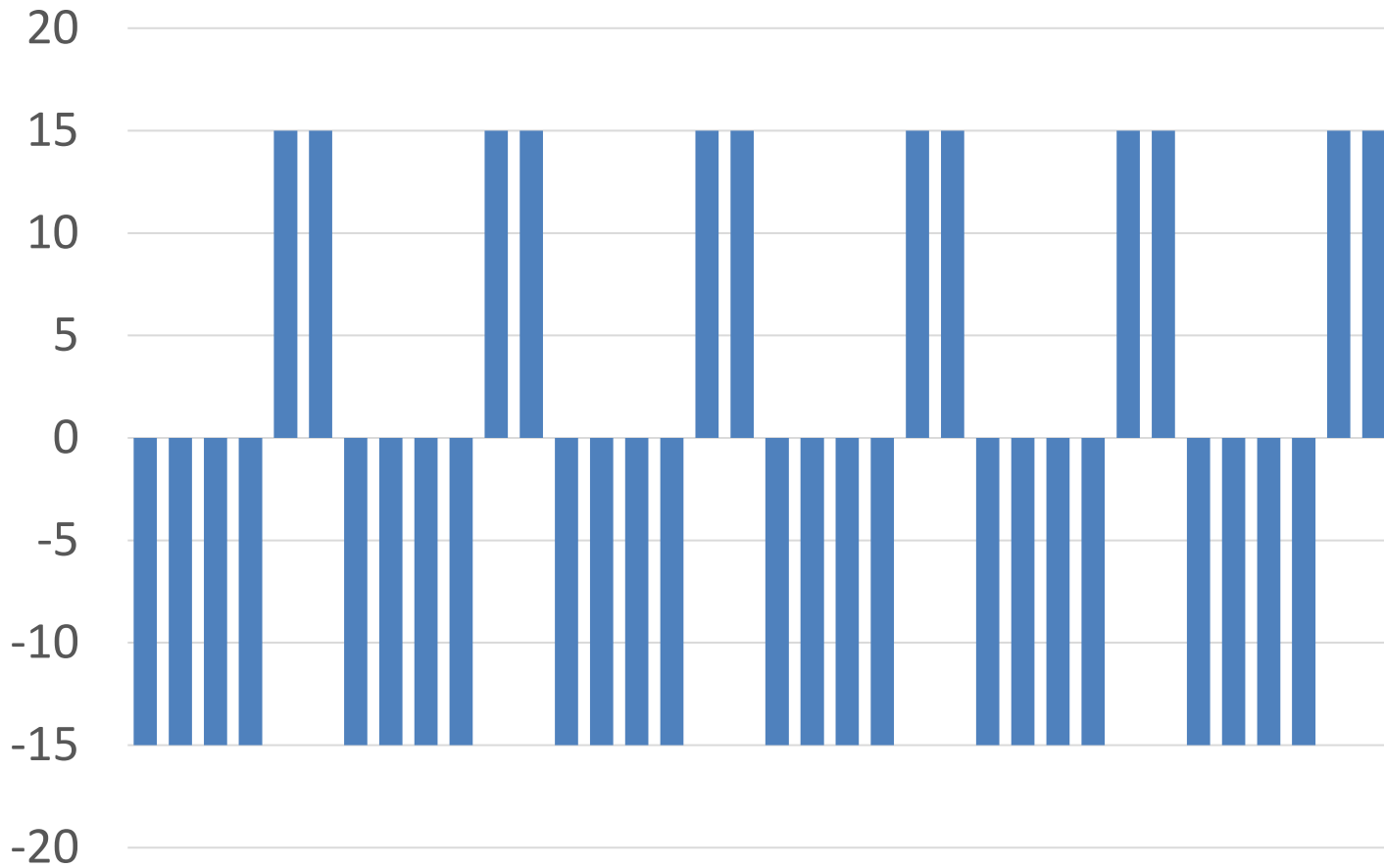
1. Since an empty argv was treated as a usage error, you were free to assert  $n > 0$  in your `MostPositiveSubsequence()` function.
2. If all the elements are negative, the result is the single least negative element.
3. If two subsequences have identical sums, the shortest, leftmost is preferred.
4. Also, performance does matter. You had to be within 20% of my benchmark.

# Why shortest first?

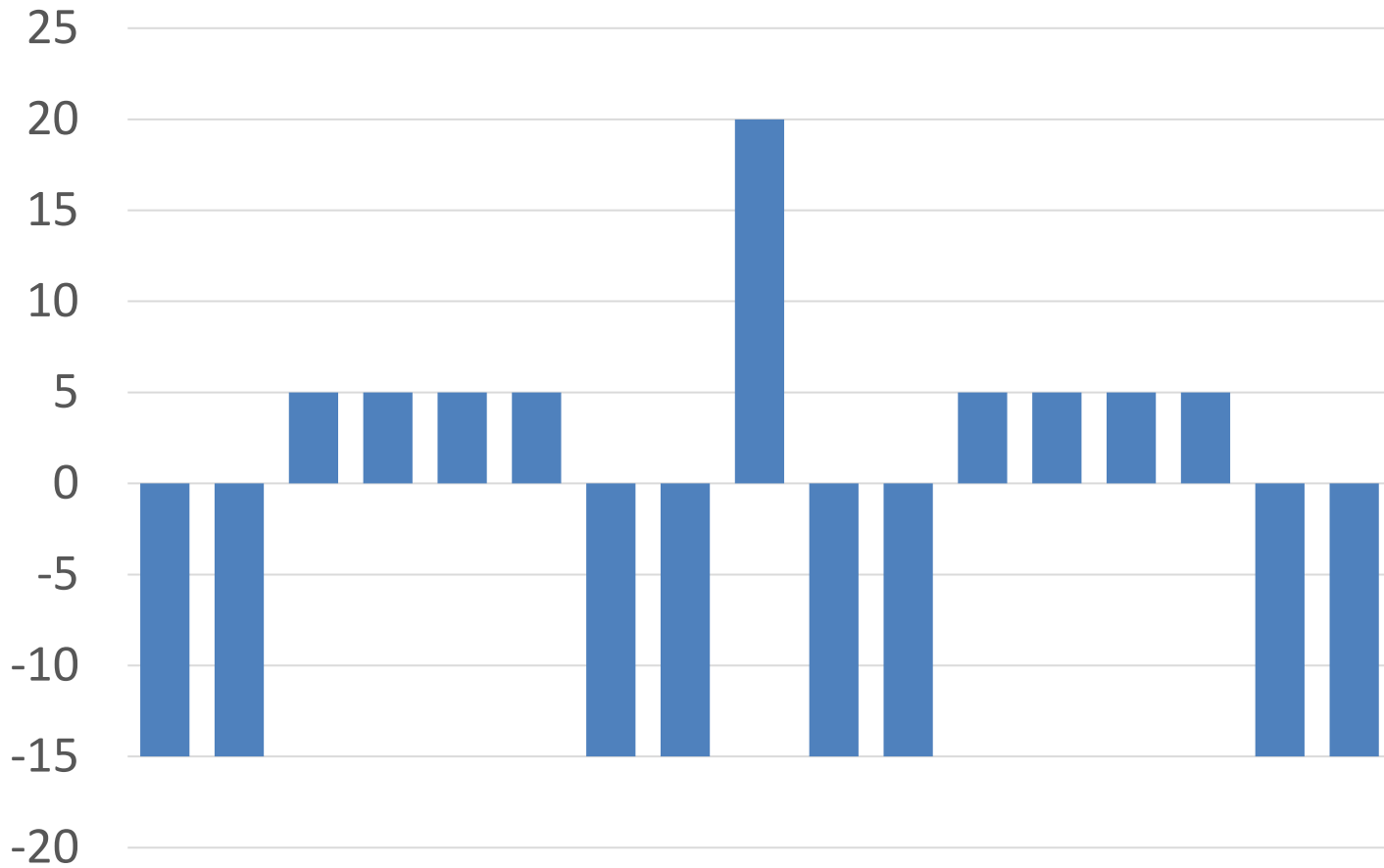
Consider this as a signal processing problem, of picking the peak.

Two common characteristics of signals are that they may have harmonics and they may repeat.

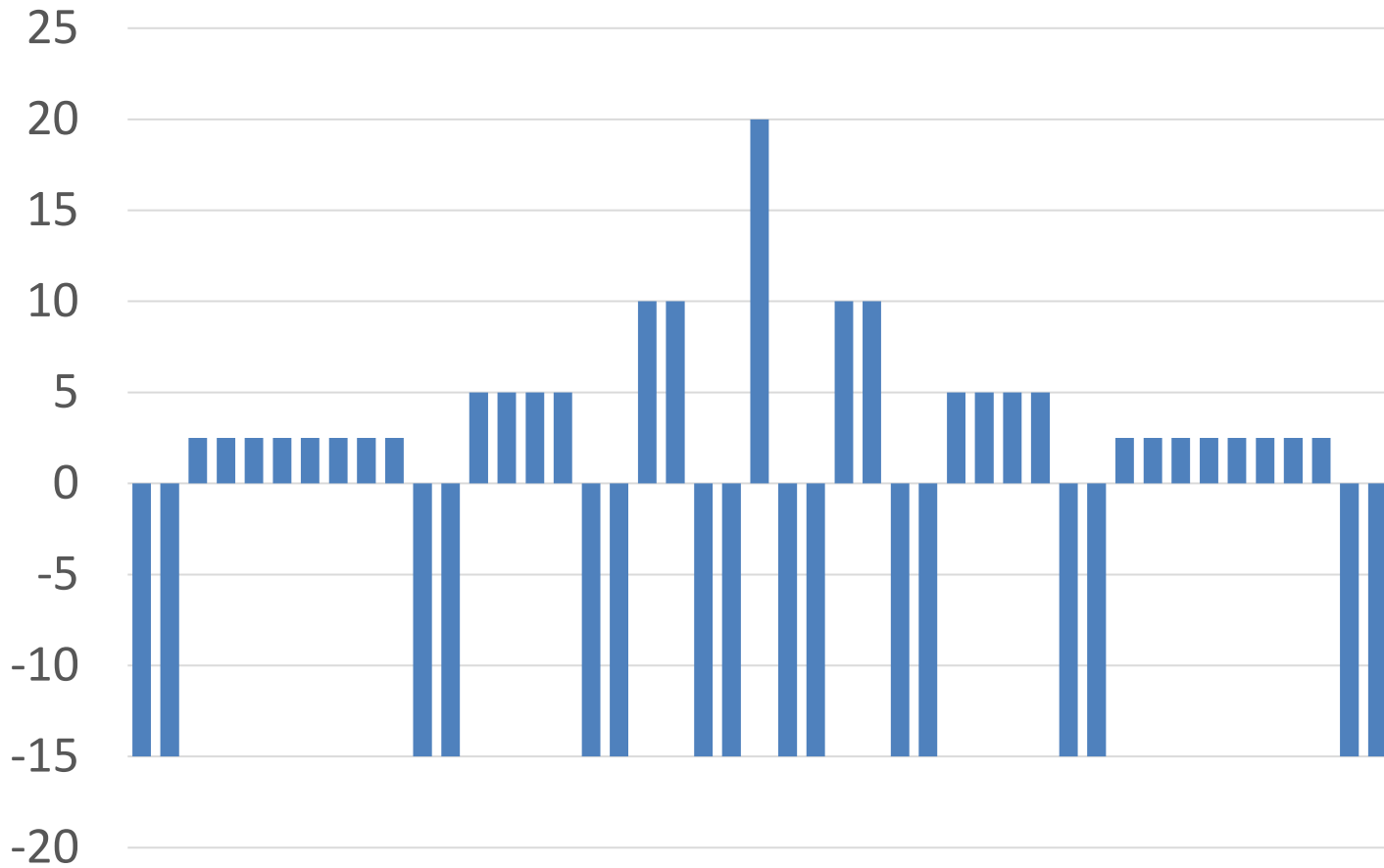
What should you choose?



What should you choose?



What should you choose?



# Naïve solution

```
int NaiveSolution( int a[ ], int n, int &bestL, int &bestR )
{
    assert( n );

    int sum, bestSum, L, R;
    bestSum = a[ 0 ];
    bestL = bestR = 0;

    for ( L = 0; L < n; L++ )
        for ( R = L, sum = 0; R < n; R++ )
            {
                sum += a[ R ];
                if ( sum > bestSum )
                    {
                        bestL = L;
                        bestR = R;
                        bestSum = sum;
                    }
            }

    return bestSum;
}
```

*What makes this naïve?*

# Naïve solution

```
int NaiveSolution( int a[ ], int n, int &bestL, int &bestR )
{
    assert( n );

    int sum, bestSum, L, R;
    bestSum = a[ 0 ];
    bestL = bestR = 0;

    for ( L = 0; L < n; L++ )
        for ( R = L, sum = 0; R < n; R++ )
            {
                sum += a[ R ];
                if ( sum > bestSum )
                    {
                        bestL = L;
                        bestR = R;
                        bestSum = sum;
                    }
            }

    return bestSum;
}
```

*What makes this naïve?*

*Once a sum  $\leq 0$ , it can't be useful.*



```

int Improved( int a[ ], int n, int &bestL, int &bestR )
{
    assert( n );

    int sum, bestSum, L, R;
    bestSum = a[ 0 ];
    bestL = bestR = 0;

    for ( L = 0; L < n; L++ )
        for ( R = L, sum = 0; R < n; R++ )
            {
                sum += a[ R ];
                if ( sum > bestSum )
                    {
                        bestL = L;
                        bestR = R;
                        bestSum = sum;
                    }
                if ( sum <= 0 )
                    break;
            }

    return bestSum;
}

```

*Stops when the sum  $\leq 0$ .  
Does not find the shortest  
sequence.*

```

int Shortest( int a[ ], int n, int &bestL, int &bestR )
{
    assert( n );

    int sum, bestSum, L, R;
    bestSum = a[ 0 ];
    bestL = bestR = 0;

    for ( L = 0; L < n; L++ )
        for ( R = L, sum = 0; R < n; R++ )
            {
                sum += a[ R ];
                if ( sum > bestSum ||
                    sum == bestSum && R - L < bestR - bestL )
                    {
                        bestL = L;
                        bestR = R;
                        bestSum = sum;
                    }
                if ( sum <= 0 )
                    break;
            }

    return bestSum;
}

```

*Finds the shortest sequence.*

```

int Shortest( int a[ ], int n, int &bestL, int &bestR )
{
    assert( n );

    int sum, bestSum, L, R;
    bestSum = a[ 0 ];
    bestL = bestR = 0;

    for ( L = 0; L < n; L++ )
        for ( R = L, sum = 0; R < n; R++ )
            {
                sum += a[ R ];
                if ( sum > bestSum ||
                    sum == bestSum && R - L < bestR - bestL )
                    {
                        bestL = L;
                        bestR = R;
                        bestSum = sum;
                    }
                if ( sum <= 0 )
                    break;
            }

    return bestSum;
}

```

*Do you like this solution?  
Anything you think is ugly?*

```

int Shortest( int a[ ], int n, int &bestL, int &bestR )
{
    assert( n );

    int sum, bestSum, L, R;
    bestSum = a[ 0 ];
    bestL = bestR = 0;

```

*Do we really need two nested loops?*

```

    for ( L = 0; L < n; L++ )
        for ( R = L, sum = 0; R < n; R++ )
            {
                sum += a[ R ];
                if ( sum > bestSum ||
                    sum == bestSum && R - L < bestR - bestL )
                    {
                        bestL = L;
                        bestR = R;
                        bestSum = sum;
                    }
            }

```

```

        if ( sum <= 0 )
            break;

```

*Two exits to the for loop.*

```

return bestSum;
}

```

```

int AlternateSolution( int a[ ], int n, int &bestL, int &bestR )
{
    assert( n );
    int sum, bestSum, L, R;
    bestSum = a[ 0 ];
    bestL = bestR = 0;

    for ( L = R = sum = 0; R < n; R++ )
    {
        sum += a[ R ];
        if ( sum > bestSum ||
            sum == bestSum && R - L < bestR - bestL )
        {
            bestL = L;
            bestR = R;
            bestSum = sum;
        }
        if ( sum <= 0 )
        {
            sum = 0;
            L = R + 1;
        }
    }
    return bestSum;
}

```

*Can this be optimized?*

```

int OptimizedAlternate( int a[ ], int n, int &bestL, int &bestR )
{
    assert( n );
    int sum, bestSum, L, R;
    bestSum = a[ 0 ];
    bestL = bestR = 0;

    for ( L = R = sum = 0; R < n; )
    {
        sum += a[ R ];
        if ( sum > bestSum ||
            sum == bestSum && R - L < bestR - bestL )
        {
            bestL = L;
            bestR = R;
            bestSum = sum;
        }
        R++;
        if ( sum <= 0 )
        {
            sum = 0;
            L = R;
        }
    }
    return bestSum;
}

```

*Can this be optimized?  
Might move the increment  
operation.*

# Agenda

1. Course details.
2. HW1 MostPositiveSubsequence( ).
3. **History of the web.**
4. Search basics.
5. Your project.

# Before the web

- 1970s Internal networks, mostly for email. Dial in for time sharing. 300 baud modems.
- 1980s USENET newsgroups. Posts were passed by polling from one machine to the next. 2400 baud modems.
- 1990s Bulletin boards, CompuServe, BIX, many others. 9600 baud modems.



Newsgroups: rec.humor  
Path:  
gmd.de!xlink.net!sol.ctr.columbia.edu!news.kei.com!ub!acsu.buffalo.edu!ubvms.cc  
.buffalo.edu!v140pxgt  
From: v140...@ubvms.cc.buffalo.edu (Daniel B Case)  
Subject: Canonical List of Light Bulb Jokes  
Message-ID: <CEr2F0.521@acsu.buffalo.edu>  
News-Software: VAX/VMS VNEWS 1.41  
Sender: nn...@acsu.buffalo.edu  
Nntp-Posting-Host: ubvmsd.cc.buffalo.edu  
Organization: University at Buffalo  
Date: Mon, 11 Oct 1993 20:34:00 GMT  
Lines: 1021

Sorry for the delay this month, but I had to clear some disk space to post this (In the future, whenever I get there, I will be using Unix so this won't happen-I hope). Anyway, here's the latest version.

⋮

<https://groups.google.com/forum/?hl=en#!original/rec.humor/uRVacQrozNs/Qrmmizp8bUUJ>

Newsgroups: rec.humor

Path:

gmd.de!xlink.net!sol.ctr.columbia.edu!news.kei.com!ub!acsu.buffalo.edu!ubvms.cc  
.buffalo.edu!v140pxgt

From: v140...@ubvms.cc.buffalo.edu (Daniel B Case)

Subject: Canonical List of Light Bulb Jokes

Message-ID: <CEr2F0.521@acsu.buffalo.edu>

News-Software: VAX/VMS VNEWS 1.41

Sender: nn...@acsu.buffalo.edu

Nntp-Posting-Host: ubvmsd.cc.buffalo.edu

Organization: University at Buffalo

Date: Mon, 11 Oct 1993 20:34:00 GMT

Lines: 1021

Sorry for the delay this month, but I had to clear some disk space to post this (In the future, whenever I get there, I will be using Unix so this won't happen-I hope). Anyway, here's the latest version.

⋮

<https://groups.google.com/forum/?hl=en#!original/rec.humor/uRVacQrozNs/Qrmmizp8bUUJ>

Newsgroups: rec.humor  
Path:  
gmd.de!xlink.net!sol.ctr.columbia.edu!news.kei.com!ub!acsu.buffalo.edu!ubvms.cc  
.buffalo.edu!v140pxgt  
From: v140...@ubvms.cc.buffalo.edu (Daniel B Case)  
Subject: Canonical List of Light Bulb Jokes  
Message-ID: <CER2F0.521@acsu.buffalo.edu>  
News-Software: VAX/VMS VNEWS 1.41  
Sender: nn...@acsu.buffalo.edu  
Nntp-Posting-Host: ubvmsd.cc.buffalo.edu  
Organization: University at Buffalo  
Date: Mon, 11 Oct 1993 20:34:00 GMT  
Lines: 1021

Sorry for the delay this month, but I had to clear some disk space to post this (In the future, whenever I get there, I will be using Unix so this won't happen-I hope). Anyway, here's the latest version.

⋮

<https://groups.google.com/forum/?hl=en#!original/rec.humor/uRVacQrozNs/Qrmmizp8bUUJ>

# The web is born

Aug 6, 1991, the WWW goes live.

1993 Mosaic browser introduced.

1994, WebCrawler and Lycos go live.



Search the web and show  for  results

Example: "Alien Abduction" UFO Roswell [Search tips](#)

**WebCrawler SELECT** [Nobody covers sports like SportsLine. Click here.](#)

► Choose one of these categories:

- [Arts & Literature](#) - [Business](#) - [Chat](#) - [Computers](#) - [Daily News](#)
- [Education](#) - [Entertainment](#) - [Government](#) - [Health & Medicine](#)
- [Internet](#) - [Kids & Families](#) - [Life & Culture](#) - [Personal Finance](#)
- [Recreation](#) - [Reference Desk](#) - [Science](#) - [Sports](#) - [Travel](#)

Get the latest dirt in our new [Gossip](#) section!

Search · [Browse](#) · [Special](#) · [Add URL](#) · [Help](#)

Copyright © 1996 [America Online, Inc.](#)  
[Disclaimer](#)



# Web surfing is born

Through the 90s, individuals and companies post pages with quirky links.

Web surfing begins.



*World's most powerful tools!*  
**Hamilton Laboratories**

Hamilton Laboratories is an independent software vendor offering tools and training to professional developers world-wide. Our main product is [Hamilton C shell](#), a tools package that recreates the original UNIX C shell and utilities completely from scratch on Windows NT, Windows 95 and OS/2, adding numerous enhancements. We also offer an on-site course, [Win32 for UNIX Programmers](#), for experienced programmers moving from UNIX to Windows NT and Windows 95.

From this site, you'll be able to learn more about [our company](#) and products and our commitment to quality, support and customer satisfaction. You can download [free updates](#) and [free demo versions](#) of our software, browse our [on-line documentation](#) and you can explore [links](#) to other resources on the Internet we hope you'll enjoy. If you have questions, please [email](#) us; we do try to answer most questions within one or two business days.

**Hamilton Laboratories**

21 Shadow Oak Drive, Sudbury, MA 01776-3165, U.S.A.  
Phone 508-440-8307 | Fax 508-440-8308 | Email [hamilton@hamiltonlabs.com](mailto:hamilton@hamiltonlabs.com)



## Links

We've assembled here links to quite a number of interesting sites we've discovered. As you can see, links on this page are *not* all purely business-related. For example, the kids' links are here simply because we have children (as many of you do also) and finding good clean sites you can safely let your kids explore is not always all that easy. If you have some recommendations for additions or if you discover any of these pages have disappeared, please [let us know](#).

Links are broken up into the following categories:

- [Geek Stuff](#)
- [Global Positioning Systems](#)
- [Humor](#)
- [Kids' Stuff](#)
- [News Organizations](#)
- [Search Engines](#)
- [Software Sites](#)
- [Spam and Privacy Issues](#)
- [Stock Market](#)
- [Weather](#)
- [Web Page Creation](#)
- [WinNT Information](#)

(This page was created from our own Internet Explorer Favorites list with [a C shell script](#).)

### Geek Stuff:

Almost everything you ever wanted to know is out there somewhere on the internet. There's also a lot of stuff you really didn't care to know. You decide.



## Humor:

The title says it all, though we admit that not all the sites we list were actually intended as humorous. Sometimes, that's the whole reason they're so funny.

[\\$95.093](#)

[About The True Mirror](#)

[Application to be Sarang's Girlfriend](#)

[Bill Gates Personal Wealth Clock](#)

[Bill Gates for President 1996!](#)

[Canonical Light Bulb Jokes](#)

[Daily Deaffirmations](#)

[Dave Letterman's Top Ten Lists](#)

[Dead People Server](#)

[Dear Abby](#)

[Dilbert Newsletter](#)

[Geek Site of the Day](#)

[IBM buys Episcopal Church](#)

[Kevin Kelm's Virtual Vomit](#)

[Light Bulb Jokes for Programmers](#)

[Listing of Bondage Stories](#)

[Make Dogs Fast!!](#)

[Microsnot Corporation](#)

[Microsoft Bids to Acquire Catholic Church](#)

[Microsoft Light Bulb Jokes](#)

[Mystical Smoking Head of 'Bob'](#)

[News of the Weird Archives](#)

[Official Outhouses of America Tour](#)

[Other Crackpot Religions!](#)

[Ovi's World of the Bizarre](#)

[Pizza Server!](#)

[Poker at the Dead Eye Saloon](#)

# Yahoo!

Mar 2, 1995, Yahoo! goes live but their links are all hand-curated.



[Exploring Mars](#)

The New York Times  
**Win a \$20,000 Trip!** **CLICK HERE!**

Looking for a [Car?](#) [Job?](#) [Date?](#)

 Search [options](#)

[Yellow Pages](#) - [People Search](#) - [Maps](#) - [Classifieds](#) - [News](#) - [Stock Quotes](#) - [Sports Scores](#)

- [Arts and Humanities](#)  
[Architecture](#), [Photography](#), [Literature](#)...
- [Business and Economy \[Xtra!\]](#)  
[Companies](#), [Investing](#), [Employment](#)...
- [Computers and Internet \[Xtra!\]](#)  
[Internet](#), [WWW](#), [Software](#), [Multimedia](#)...
- [Education](#)  
[Universities](#), [K-12](#), [College Entrance](#)...
- [Entertainment \[Xtra!\]](#)  
[Cool Links](#), [Movies](#), [Music](#), [Humor](#)...
- [Government](#)  
[Military](#), [Politics \[Xtra!\]](#), [Law](#), [Taxes](#)...
- [Health \[Xtra!\]](#)  
[Medicine](#), [Drugs](#), [Diseases](#), [Fitness](#)...
- [News and Media \[Xtra!\]](#)  
[Current Events](#), [Magazines](#), [TV](#), [Newspapers](#)...
- [Recreation and Sports \[Xtra!\]](#)  
[Sports](#), [Games](#), [Travel](#), [Autos](#), [Outdoors](#)...
- [Reference](#)  
[Libraries](#), [Dictionaries](#), [Phone Numbers](#)...
- [Regional](#)  
[Countries](#), [Regions](#), [U.S. States](#)...
- [Science](#)  
[CS](#), [Biology](#), [Astronomy](#), [Engineering](#)...
- [Social Science](#)  
[Anthropology](#), [Sociology](#), [Economics](#)...
- [Society and Culture](#)  
[People](#), [Environment](#), [Religion](#)...

[My Yahoo!](#) - [Yahooligans! for Kids](#) - [Beatrice's Web Guide](#) - [Yahoo! Internet Life](#)  
[Weekly Picks](#) - [Today's Web Events](#) - [Chat](#) - [Weather Forecasts](#)  
[Random Yahoo! Link](#) - [Yahoo! Shop](#)

National Yahoos [Canada](#) - [France](#) - [Germany](#) - [Japan](#) - [U.K. & Ireland](#)

# Yahoo!

Jerry Yang and David Filo, both PhD students in engineering at Stanford, start Jerry's Guide to the Web.

Jerry hated the name, so it became Yahoo!, a meaningless acronym.

Hand-crafted directory to the web.

Jerry and David categorized 1000 sites/day.

Initially running out of a trailer on campus.

Moved to free hosting on Netscape's machines in early 1995.

*"It was a labor love – lots of labor, since no software program could evaluate and categorize sites."*

# AltaVista

Dec 15, 1995, DEC Research's AltaVista goes live with the first modern search engine architecture.

Used index stream readers invented by Mike Burrows.

AltaVista quickly becomes most popular engine.

Bought by Yahoo! in 2003.



Try your search in: [Shopping](#) • [Images](#) • [Video](#) • [MP3/Audio](#) • [News](#) • [Autos](#) • [Tech](#)

Search for:

[Help](#) | [Customize Settings](#) | [Family Filter is off](#)

any language ▾

Search

[Search Assistant](#) | [Advanced Search](#)

**Shopping:** [Compare Prices](#) • [Local Deals & Coupons](#) • [Web Deals & Rebates](#) • [uBid Auction](#)

**Tools:** [Email](#) • [Translate](#) • [Maps](#) • [Directions](#) • [Yellow Pages](#) • [People Finder](#) • [Find A Date](#)  
[Find Downloads](#) • [Text-Only Search](#) • [Weight Calculator](#) • [Find A Job](#) • [Find A Home](#) • [More...](#)

**News:** [Supreme Court Upholds Use of Force in Guarding...](#) • [More News...](#)

[Web Site Hosting](#) • [Insurance Quotes](#) • [Radio Pet Fence](#) • [Get Cash Back](#) • [Buy A Computer](#)  
[Vacation Plans](#) • [Online Casinos & Gambling](#) • [DVD Players](#) • [NBA Tickets](#) • [Win Free Travel](#)

**Arts & Entertainment**

[Culture](#), [Celebrities](#), [Movies...](#)

**Music**

[Artists](#), [Genres](#), [MP3...](#)

**Business Center**

[Internet Search Services](#)

[AltaVista Enterprise Software](#)

[Download a trial version now](#)

[Submit A Site](#)

[List Your Products](#)

[Advertise With Us](#)

[Access Files From Anywhere](#)

**Autos**

[Buy & Sell](#), [Guides](#), [Repair...](#)

**People & Chat**

[Chat](#), [Email](#), [Personals...](#)

**Computing**

[Hardware](#), [Internet](#), [Software...](#)

**Personal**

[Family](#), [Intimacy](#), [Home...](#)

**Games**

[Gambling](#), [Role Playing](#), [Video...](#)

**Travel**

[Activities](#), [Destinations](#), [Trips...](#)

**Health & Fitness**

[Conditions](#), [Medicine](#), [Insurance...](#)

**Shopping**

[Coupons](#), [Deals](#), [Wireless...](#)

# Google

Sep 4, 1998, Google goes live with PageRank, named after Larry Page, coinvented by Sergey Brin.

Introduced a much simpler page where they promised that all their results were algorithmic, none paid and that they would do no evil.

Quickly became the dominant engine.



Search 1,326,920,000 web pages

[Advanced Search](#)  
[Preferences](#)

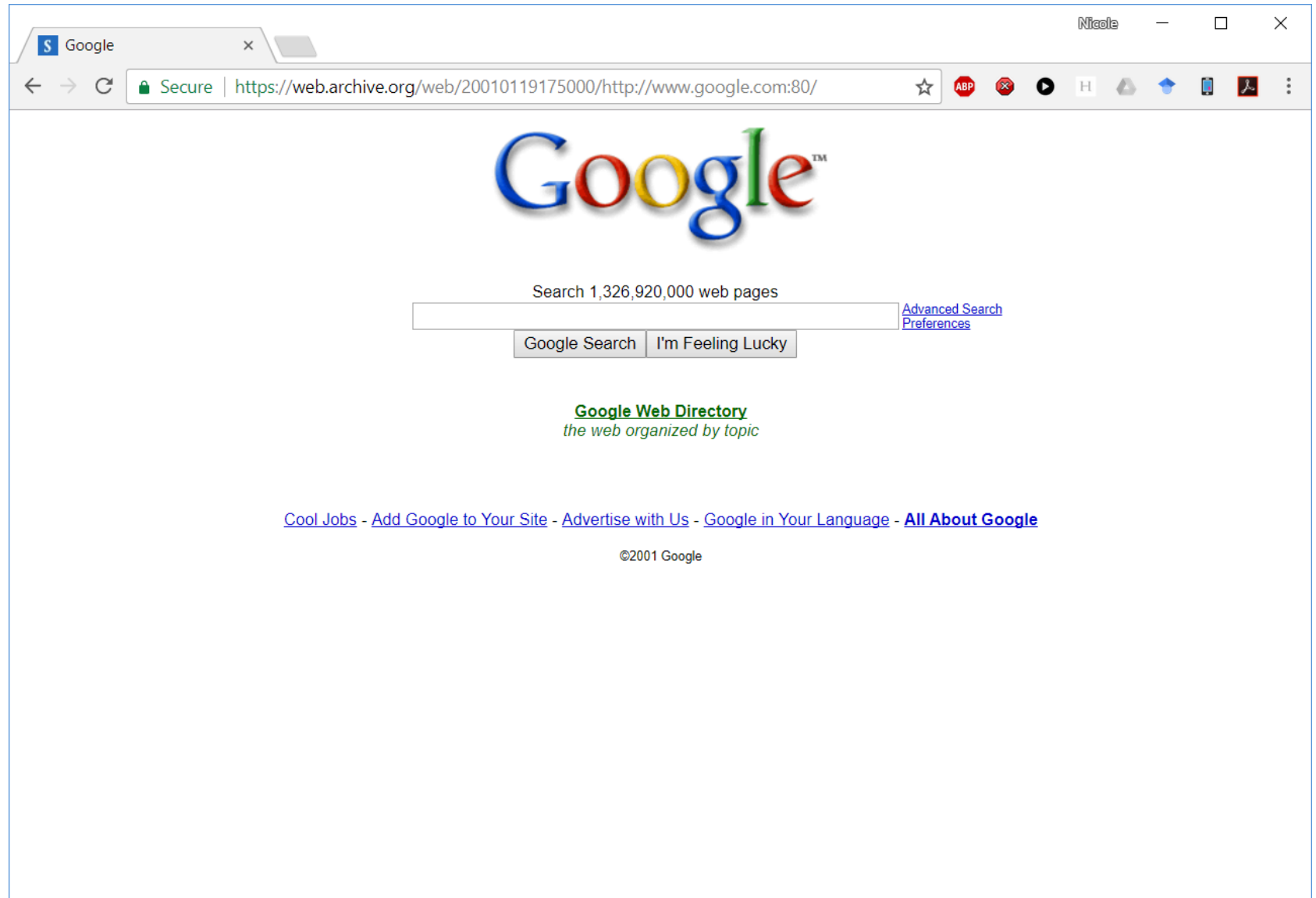
Google Search I'm Feeling Lucky

**Google Web Directory**  
*the web organized by topic*

[Cool Jobs](#) - [Add Google to Your Site](#) - [Advertise with Us](#) - [Google in Your Language](#) - [All About Google](#)



# Why did this become dominant?



# Bing

Jan 2005, Microsoft's own engine goes live, initially as MSN Search.



**msn.** Search

[Web](#) [News](#) [Images](#) [Desktop](#) [Encarta](#)

[+Search Builder](#) [Settings](#) [Help](#) [Español](#)

It's a lot easier to do something if you know it's possible.

In business, this called either an *imitator* or *follower* strategy.

In engineering, if you know something's possible, you know not to give up.

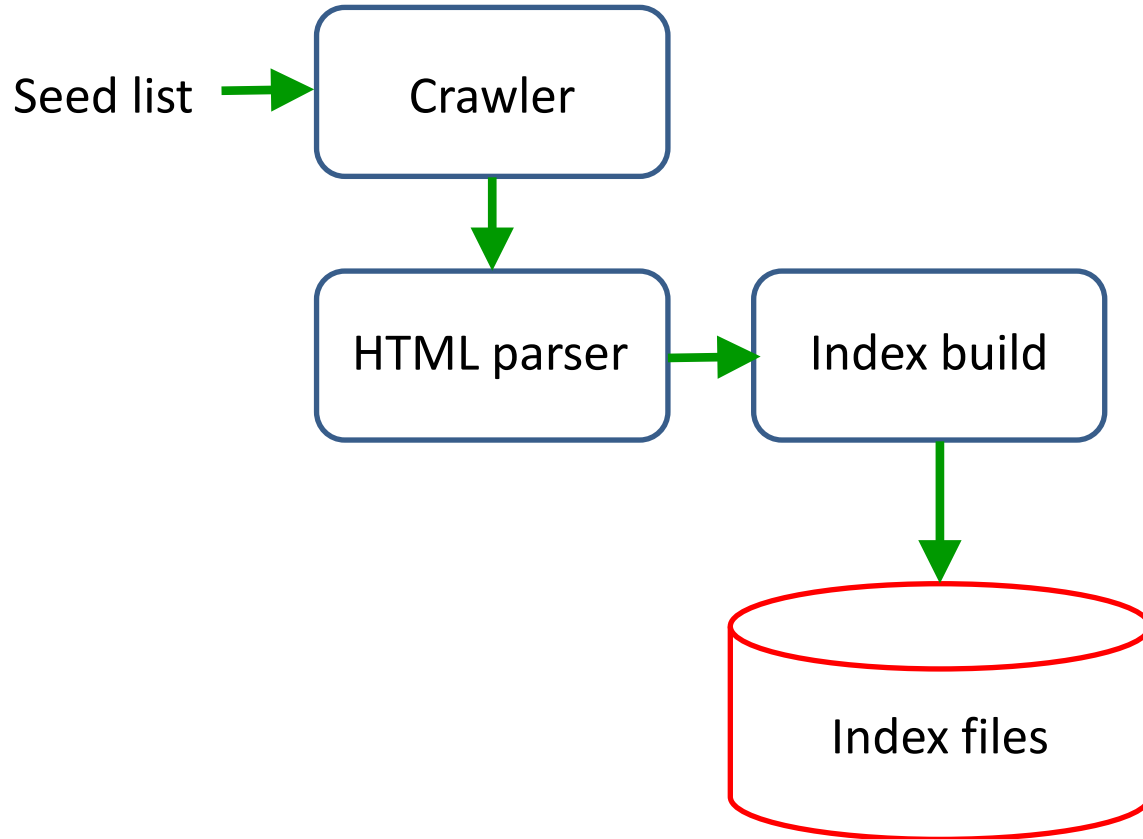
# Agenda

1. Course details.
2. HW1 MostPositiveSubsequence( ).
3. History of the web.
4. **Search basics.**
5. Your project.

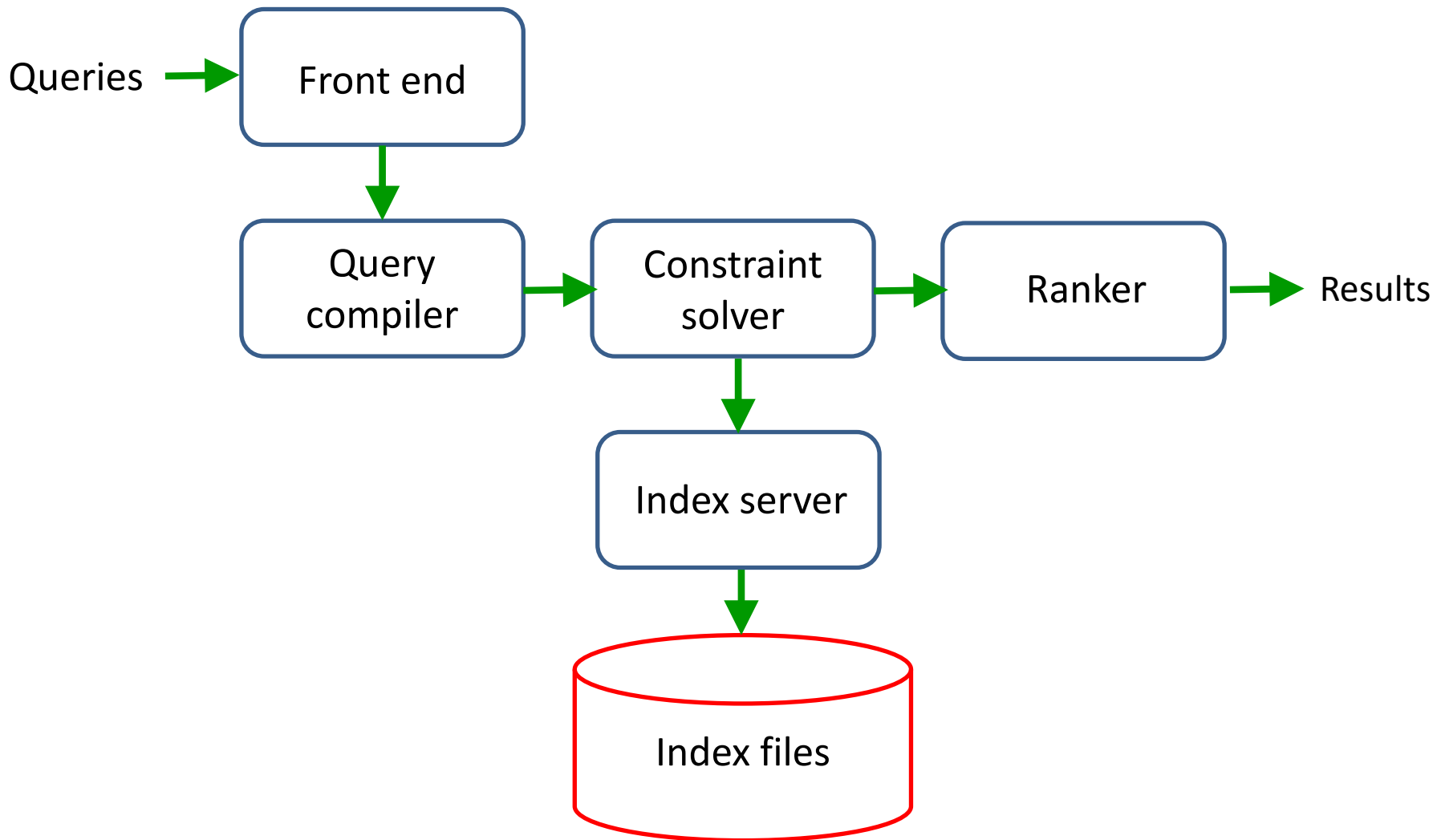
# The basic parts to a search engine

1. HTML parser.
2. Crawler.
3. Index.
4. Constraint solver.
5. Query language.
6. Ranker.
7. Front end.

# The index build side

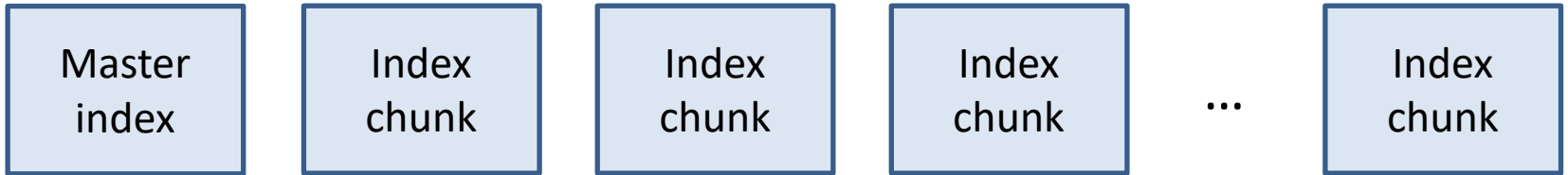


# The query serve side

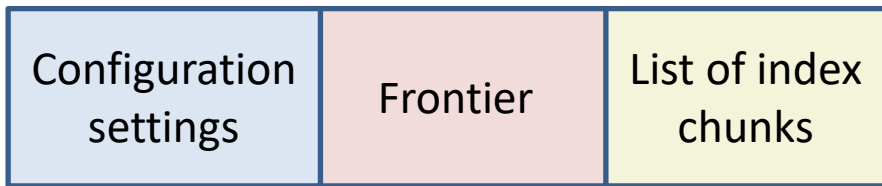




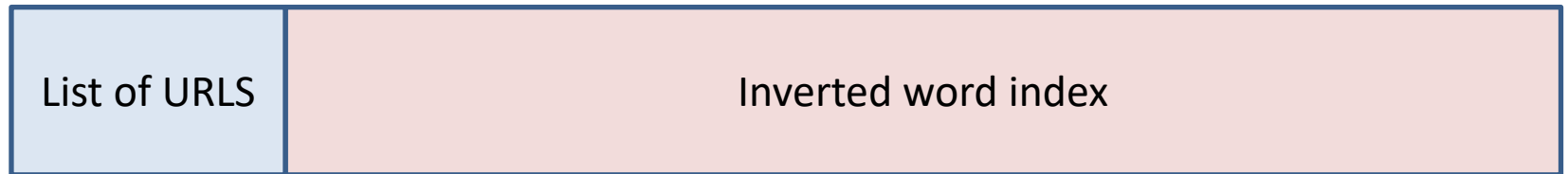
A search engine index is typically a set of files



A master index



Each index chunk



# HTML Parser

Extract the content from a HTML file as a series of tokens in the title and the body of the document and a set of links with anchor text to other documents.

# HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
  <meta content="text/html; charset=utf-8" http-equiv="Content-Type" />

  <title>Page title shown in the browser tab</title>
  <script>Lots of gibberish</script>

  <link href="MyStyles.css" rel="stylesheet" type="text/css" />
  <link href="https://mydomain/favicon.ico" rel="shortcut icon" />

  <meta name="DC.Rights" content="Copyright 2018 my name" />
  <meta name="description" content="Short abstract." />
  <meta name="keywords" content="arbitrary list of words" />
</head>

<body>
:
</body>
</html>
```

# HTML Parser

Realistically, many teams find getting the HTML parser to work is the most time-consuming part of the project.

There is so much bad, broken HTML out there on the web that most teams struggle to keep their parsers from crashing.

# HTML Parser

Realistically, many teams find getting the HTML parser to work is the most time-consuming part of the project.

There is so much bad, broken HTML out there on the web that most teams struggle to keep their parsers from crashing.

This is a big part of the *surface area* of your project.

# HTML Parser

Common pitfall: Overengineering and overcomplexity.

Think carefully: What does it need to do?

Does it need to parse the entire document structure?

HW3 will try to point you in the right direction.

# Crawler

1. Manage a frontier of new links to be crawled.
2. Decide what will or will not be crawled and in what order.
3. Keep track of what's already been crawled.
4. Read pages over HTTP and HTTPS.
5. Obey robots.txt files.
6. Deal with redirects.

All of this has to be highly multithreaded so you don't wait on slow sites, instead overlapping them.

You will also want to spread it across multiple machines and will need to decide how to divide it up.

This also is part of your *surface area*.

# Crawler

Typically maintains pool of worker processes or threads to read and parse webpages.

Each worker retrieves the file and queues it for the HTML parser which creates an object.

Links go into the frontier, perhaps with anchor.

Words go into the index.



# Crawler

It's very easy to DOS a site with thousands of threads on a bunch of AWS machines with gigabit connections.

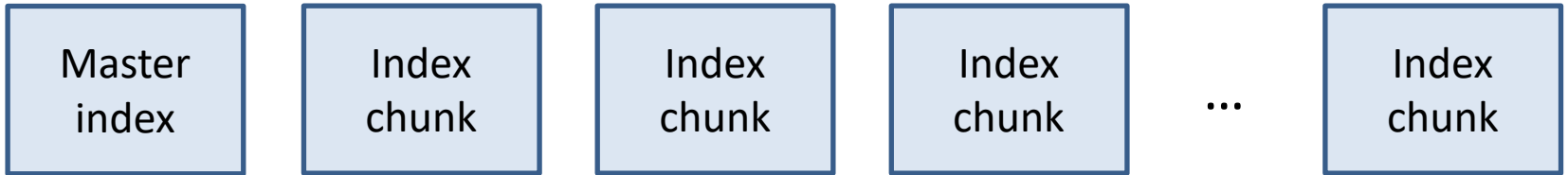
So, we'll talk about politeness and putting your contact info in your *User-Agent:* field so complaints go to you, not me.

# Index

A merged *inverted word index* of all the documents that have been crawled, allowing you to report all the documents and individual locations (postings) where any given word was found.

Due to the size, the posting lists will have to be on disk but you'll map them right into your process memory space for performance.

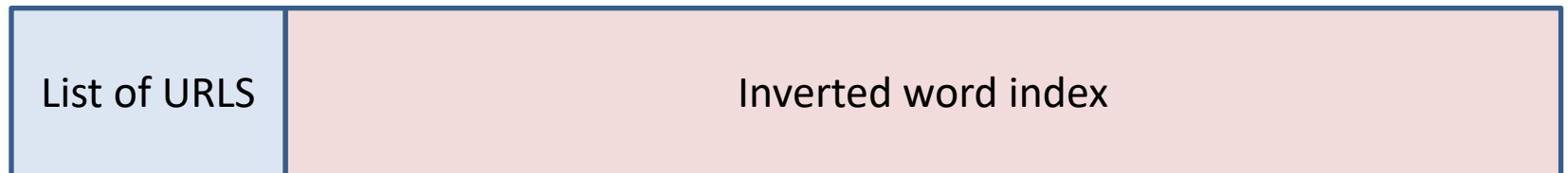
A search engine index is typically a set of files



A master index



Each index chunk



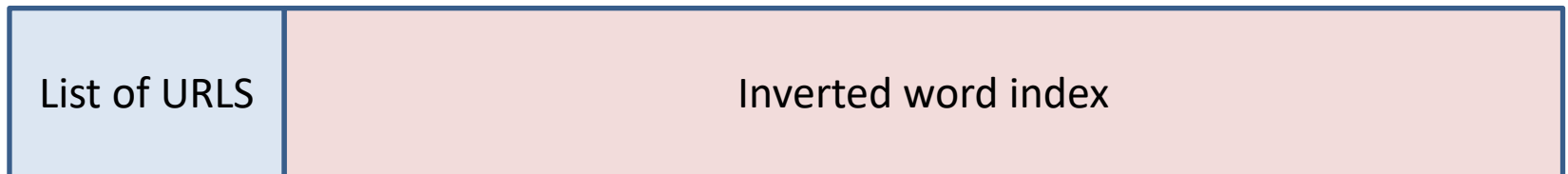
The inverted word index operates list the index at the back of a book.

And index in a book lists all pages where a word or topic appears.

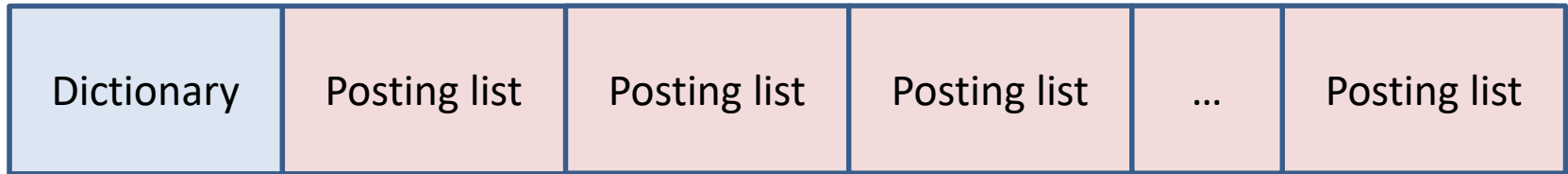
If you want to know the list of pages where some combination of topics appears, all you need to do is form the appropriate union or intersection of the lists.

It's called inverted because we think of the original text as the non-inverted form.

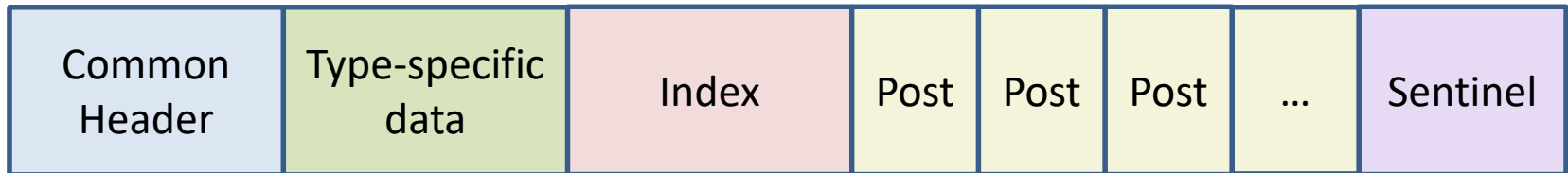
Each index chunk



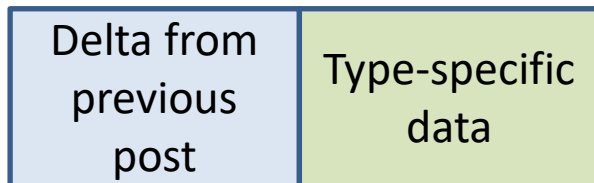
## The inverted word index



## A posting list



## An individual post



# Index Stream Reader (ISR)

Finds the next occurrence of the desired token or combination of child ISRs.

Individual words,

Document ends,

Or'ing or And'ding of term,

Phrases, and

Notes

# Constraint solver

Given an inverted word index and a constraint, e.g., a list of words that must appear together or as a phrase in a document, find the list of matching documents.

## Consider these posting lists

quick	10	27	105	513	518	520
brown	28	50	62	70	514	790
fox	87	106	515	550	1200	
#DocEnd	112	570	1006	1704		

To read and merge these lists, we need to move from one entry to the next.

We'll do that with an ISR (index stream reader).

The ISR for each token has to be able to report its current location and attributes, and it needs Next( ) and Seek( ) functions.



## OR'ing streams

quick	10	27	105	513	518	520
brown	28	50	62	70	514	790
fox	87	106	515	550	1200	
#DocEnd	112	570	1006	1704		

quick   fox	10	27	87	105	106	513	515	518	520
-------------	----	----	----	-----	-----	-----	-----	-----	-----

An OR ISR simply merges the streams.

No need to pay attention to document boundaries. Each post is in whichever posting list and whatever document it happens to be.

## AND'ing streams

quick	10	27	105	513	518	520
brown	28	50	62	70	514	790
fox	87	106	515	550	1200	
#DocEnd	112	570	1006	1704		

quick fox ?

AND'ing of terms should find occurrences of all the terms within a single document.

Should it return every possible combination, every combination only changing the nearest ISR or the first match in each matching document?

# AND'ing streams

Easier to consider if we show the document boundaries.

quick	10	27	105		513	518	520			
brown	28	50	62	70	514				790	
fox	87	106			515	550				1200
#DocEnd					112			570	1006	1704

quick fox ?

To determine what document a post falls within, we advance a #DocEnd ISR to the next document end, where we can retrieve information about the document, including its length.

This tells us the start and end points of the document and whether all the word ISRs point within the same document.

# AND'ing streams

quick	10	27	105		513	518	520			
brown	28	50	62	70	514				790	
fox	87	106			515	550				1200
#DocEnd					112			570	1006	1704

quick fox *How many possible combinations?  
Can you reach all of them in a single pass, all ISRs only moving forward?*

AND'ing of terms should find occurrences of all the terms within a single document.

Should it return every possible combination, every combination only changing the nearest ISR or the first match in each matching document?

# AND'ing streams

quick	10	27	105	513	518	520			
brown	28	50	62	70	514		790		
fox	87	106			515	550		1200	
#DocEnd				112			570	1006	1704

quick fox *How many possible combinations? 6*  
*Can you reach all of them in a single pass, all ISRs only moving forward? No.*

Should it return every possible combination, every combination only changing the nearest ISR or the first match in each matching document?

# AND'ing streams

quick	10	27	105	513	518	520			
brown	28	50	62	70	514		790		
fox	87	106			515	550		1200	
#DocEnd				112			570	1006	1704

quick fox *How many possible combinations? 6*  
*Can you reach all of them in a single pass, all ISRs only moving forward? No.*

Should it return every possible combination, every combination only changing the nearest ISR or the first match in each matching document?

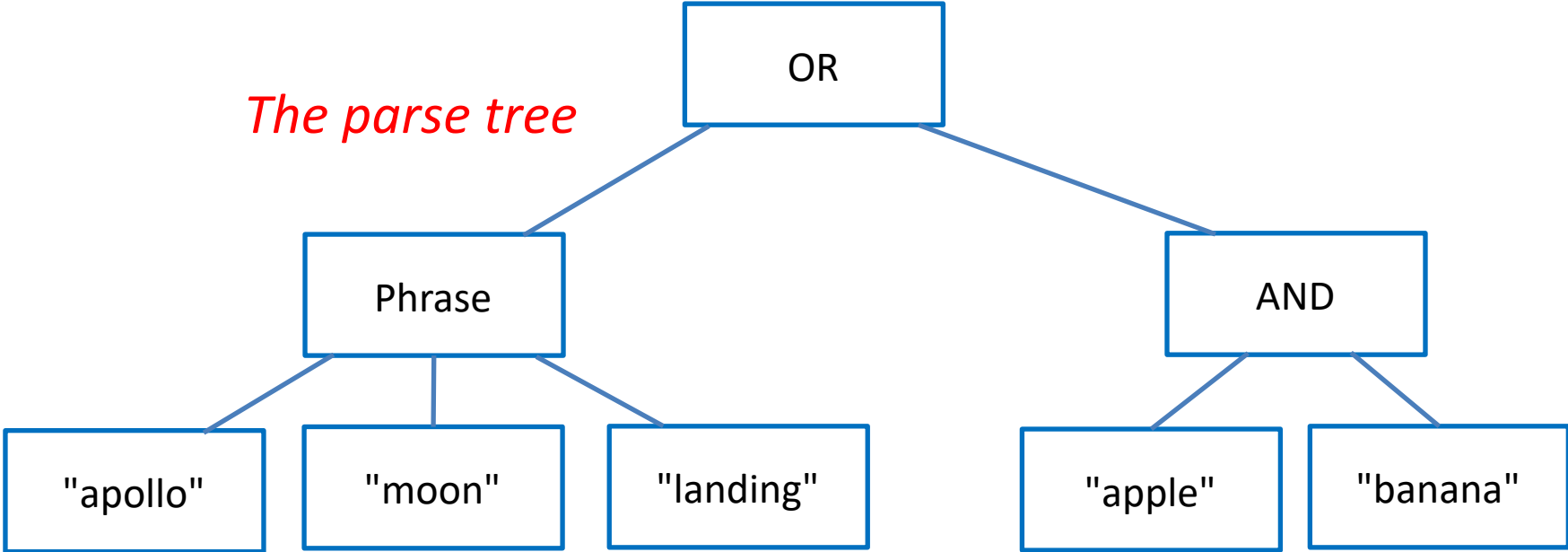
The point of the constraint solver is to find matching pages. Once any match on the page has been found, it's the ranker's job to figure out what to do next

# Query language

Compile a query as typed, e.g., with quotes around phrases, into a structure the constraint solver understands.

The query language and the ISRs will be recursive

"apollo moon landing" | ( apple banana )





# Front end

Report the results back, possibly including snippets with a simple HTTP server.

# Ranker

Use the compiled query to retrieve matching pages from the index, then score them by a statistical, heuristic or other method and pick the ten best.

# Result quality

To be useful as a search engine, it must produce good results quickly.

# Result quality

Traditional measures:

**Precision**      The fraction of the results that are relevant.

**Recall**      The fraction of all relevant results returned.

# Result quality

And because it matters that the top results be the best:

**Relevance**      An estimate of the overall quality of the page as an result for this query that can be used for *ranking* (ordering) the results.

# Kinds of rank

## Dynamic rank

How good a match this page is to this specific query.

## Static rank

How good the page is, knowing nothing about the query.

## Dynamic rank: Matching the page to the query

Traditionally methods in information retrieval considered the page as a *bag of words* and ostensibly measured the *mathematical or statistical similarity*.

For example, one method claimed to consider the *document and the query as vectors* and to calculate the *angle between them*.

# Bag of words

A scoring method that considers only that the search words occur in the document, not where or what relation to each other.



# tf-idf

Term-frequency, inverse document frequency.

Easily the most famous bag of words technique. The more occurrences of a rare word, the better.

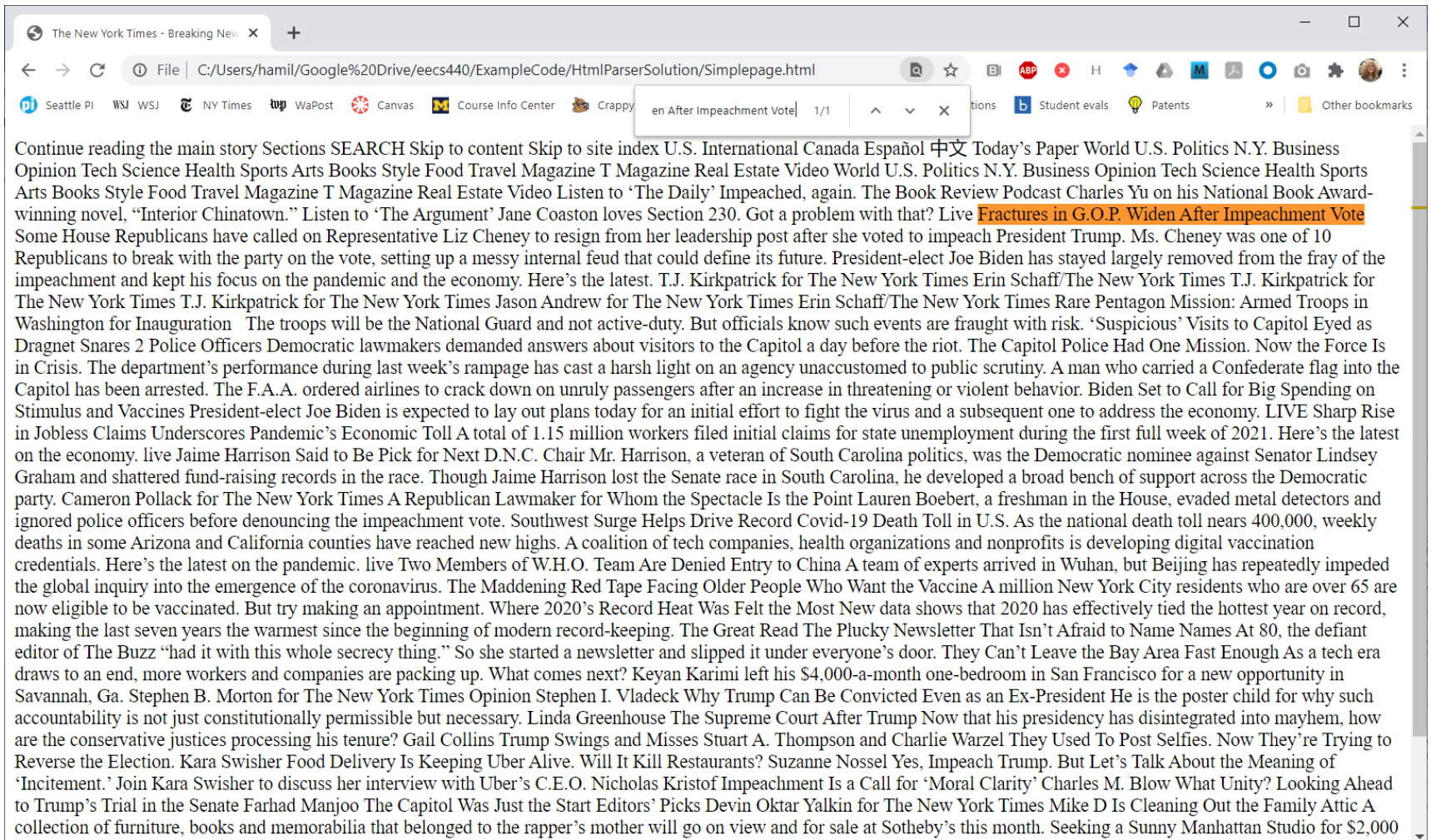
# Bag of words

Here's a sample NY Times page from Jan 25, 2021.

The screenshot shows the New York Times homepage. At the top, there's a navigation bar with the date "Monday, January 25, 2021" and the newspaper's name "The New York Times" in a large, stylized font. To the right of the name, there's a weather widget showing "29°F" and "31° 26°" and a stock market indicator for Nasdaq showing "-0.02%". Below the main title, there's a horizontal menu with various sections: World, U.S., Politics, N.Y., Business, Opinion, Tech, Science, Health, Sports, Arts, Books, Style, Food, Travel, Magazine, T Magazine, Real Estate, and Video. There are also three promotional banners for "The Daily" podcast, "The Book Review Podcast" featuring Charles Yu, and "The Argument" podcast featuring Jane Coaston. The main content area is divided into three columns. The left column has a "LIVE" tag and a headline "Fractures in G.O.P. Widen After Impeachment Vote" with a sub-headline "Some House Republicans have called on Representative Liz Cheney to resign from her leadership post after she voted to impeach President Trump." The middle column features a large photograph of the U.S. Capitol building seen through a chain-link fence, with several people standing in the foreground. The right column has a headline "'Suspicious' Visits to Capitol Eyed as Dragnet Snares 2 Police Officers" and a sub-headline "Democratic lawmakers demanded answers about visitors to the Capitol a day before the riot." At the bottom of the page, there's a URL: <https://www.nytimes.com/2021/01/08/books/review/podcast-charles-yu-interior-chinatown-david-brown-henry-adams-last-american-aristocrat.html>

# Bag of words

Here it is stripped of HTML and CSS but the text remains.

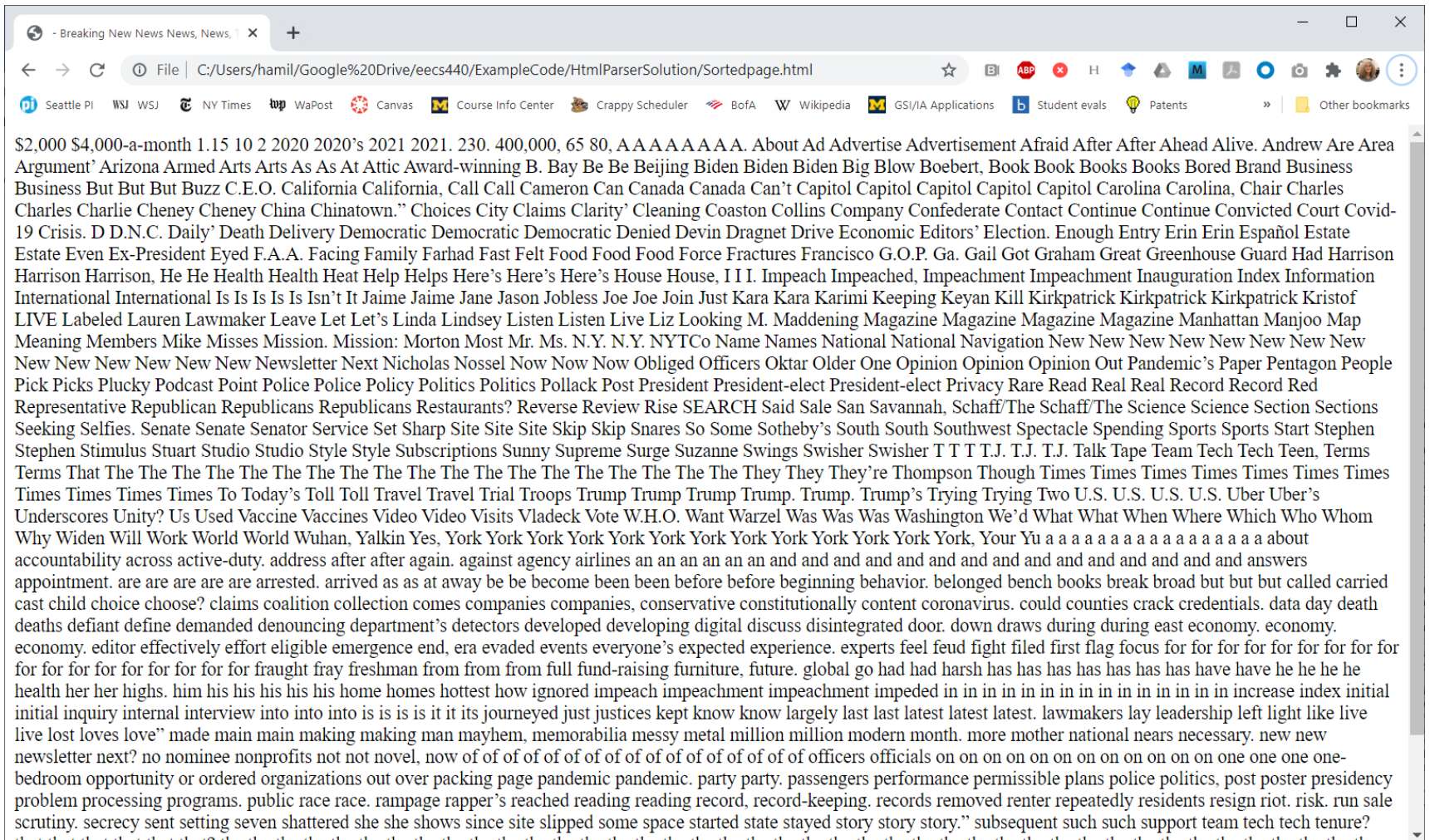


The screenshot shows a web browser window with the address bar displaying the file path: C:/Users/hamil/Google%20Drive/eecs440/ExampleCode/HtmlParserSolution/Simplepage.html. The browser's search bar contains the text "en After Impeachment Vote". Below the search bar, there is a list of text-based links and snippets, including "Continue reading the main story Sections SEARCH Skip to content Skip to site index U.S. International Canada Español 中文 Today's Paper World U.S. Politics N.Y. Business Opinion Tech Science Health Sports Arts Books Style Food Travel Magazine T Magazine Real Estate Video World U.S. Politics N.Y. Business Opinion Tech Science Health Sports Arts Books Style Food Travel Magazine T Magazine Real Estate Video Listen to 'The Daily' Impeached, again. The Book Review Podcast Charles Yu on his National Book Award-winning novel, 'Interior Chinatown.' Listen to 'The Argument' Jane Coaston loves Section 230. Got a problem with that? Live Fractures in G.O.P. Widen After Impeachment Vote Some House Republicans have called on Representative Liz Cheney to resign from her leadership post after she voted to impeach President Trump. Ms. Cheney was one of 10 Republicans to break with the party on the vote, setting up a messy internal feud that could define its future. President-elect Joe Biden has stayed largely removed from the fray of the impeachment and kept his focus on the pandemic and the economy. Here's the latest. T.J. Kirkpatrick for The New York Times Erin Schaff/The New York Times T.J. Kirkpatrick for The New York Times T.J. Kirkpatrick for The New York Times Jason Andrew for The New York Times Erin Schaff/The New York Times Rare Pentagon Mission: Armed Troops in Washington for Inauguration The troops will be the National Guard and not active-duty. But officials know such events are fraught with risk. 'Suspicious' Visits to Capitol Eyed as Dragnet Snares 2 Police Officers Democratic lawmakers demanded answers about visitors to the Capitol a day before the riot. The Capitol Police Had One Mission. Now the Force Is in Crisis. The department's performance during last week's rampage has cast a harsh light on an agency unaccustomed to public scrutiny. A man who carried a Confederate flag into the Capitol has been arrested. The F.A.A. ordered airlines to crack down on unruly passengers after an increase in threatening or violent behavior. Biden Set to Call for Big Spending on Stimulus and Vaccines President-elect Joe Biden is expected to lay out plans today for an initial effort to fight the virus and a subsequent one to address the economy. LIVE Sharp Rise in Jobless Claims Underscores Pandemic's Economic Toll A total of 1.15 million workers filed initial claims for state unemployment during the first full week of 2021. Here's the latest on the economy. live Jaime Harrison Said to Be Pick for Next D.N.C. Chair Mr. Harrison, a veteran of South Carolina politics, was the Democratic nominee against Senator Lindsey Graham and shattered fund-raising records in the race. Though Jaime Harrison lost the Senate race in South Carolina, he developed a broad bench of support across the Democratic party. Cameron Pollack for The New York Times A Republican Lawmaker for Whom the Spectacle Is the Point Lauren Boebert, a freshman in the House, evaded metal detectors and ignored police officers before denouncing the impeachment vote. Southwest Surge Helps Drive Record Covid-19 Death Toll in U.S. As the national death toll nears 400,000, weekly deaths in some Arizona and California counties have reached new highs. A coalition of tech companies, health organizations and nonprofits is developing digital vaccination credentials. Here's the latest on the pandemic. live Two Members of W.H.O. Team Are Denied Entry to China A team of experts arrived in Wuhan, but Beijing has repeatedly impeded the global inquiry into the emergence of the coronavirus. The Maddening Red Tape Facing Older People Who Want the Vaccine A million New York City residents who are over 65 are now eligible to be vaccinated. But try making an appointment. Where 2020's Record Heat Was Felt the Most New data shows that 2020 has effectively tied the hottest year on record, making the last seven years the warmest since the beginning of modern record-keeping. The Great Read The Plucky Newsletter That Isn't Afraid to Name Names At 80, the defiant editor of The Buzz "had it with this whole secrecy thing." So she started a newsletter and slipped it under everyone's door. They Can't Leave the Bay Area Fast Enough As a tech era draws to an end, more workers and companies are packing up. What comes next? Keyan Karimi left his \$4,000-a-month one-bedroom in San Francisco for a new opportunity in Savannah, Ga. Stephen B. Morton for The New York Times Opinion Stephen I. Vladeck Why Trump Can Be Convicted Even as an Ex-President He is the poster child for why such accountability is not just constitutionally permissible but necessary. Linda Greenhouse The Supreme Court After Trump Now that his presidency has disintegrated into mayhem, how are the conservative justices processing his tenure? Gail Collins Trump Swings and Misses Stuart A. Thompson and Charlie Warzel They Used to Post Selfies. Now They're Trying to Reverse the Election. Kara Swisher Food Delivery Is Keeping Uber Alive. Will It Kill Restaurants? Suzanne Nossel Yes, Impeach Trump. But Let's Talk About the Meaning of 'Incitement.' Join Kara Swisher to discuss her interview with Uber's C.E.O. Nicholas Kristof Impeachment Is a Call for 'Moral Clarity' Charles M. Blow What Unity? Looking Ahead to Trump's Trial in the Senate Farhad Manjoo The Capitol Was Just the Start Editors' Picks Devin Oktor Yalkin for The New York Times Mike D Is Cleaning Out the Family Attic A collection of furniture, books and memorabilia that belonged to the rapper's mother will go on view and for sale at Sotheby's this month. Seeking a Sunny Manhattan Studio for \$2,000



# Bag of words

Here it is with the words in the title and body sorted. Tf-idf can't tell the difference.



# At Microsoft

At Microsoft, we separated the occurrences into four metastreams, in order of importance.

1. Anchor text.
2. URL.
3. Title.
4. Body text.

# Heuristics

1. Pick the rarest word in the query, then iterate over its occurrences on the page.
2. At each place, move the pointers to the other words to as close to the correct relative position as possible.
3. Score that that set, called a *span*, counting the number of spans, the number of exact phrases, in order, close together, etc.
4. Score the counts, weighting exact phrases the title as more important than other matches.

# Heuristics

Altogether, had about 700 heuristics in a linear combination.

Tuned them using gradient descent and approximately 150K labeled pages and queries.

# What we found

Finding the search words in the right order and close together was more important than simply finding lots of them.

Added tf-idf because under pressure that of course we should consider it. Removed it because it added nothing to our relevance but slowed our crawl.

It mattered a lot where we found the matching terms.

Anchor > URL > Title > Body text



# Static rank

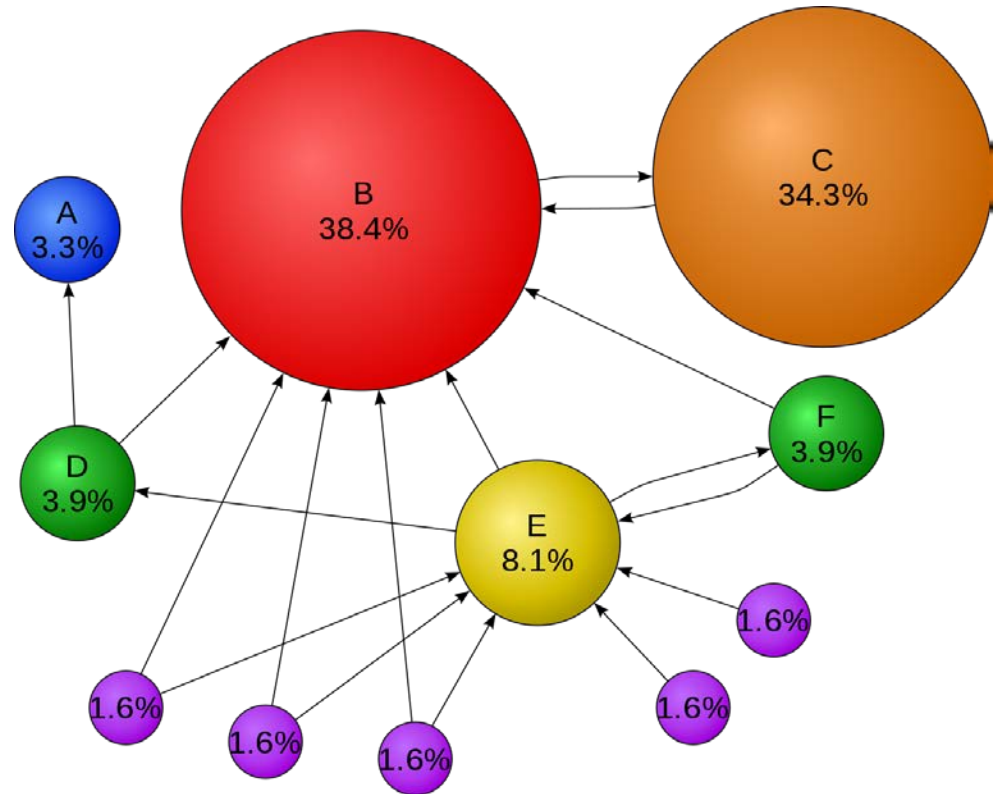
Some pages are just better than others before you know anything about the query.

# Static rank

1. Some domains are better than others, e.g., .gov or .edu over .biz.
2. Short URLs are better.
3. Short titles are probably better.
4. Some pages may be obvious spam.
5. Some pages may have lots of other pages pointing to them, e.g., PageRank.

# PageRank

The basic idea: The more and better links to a page, the more likely it should rank higher.



# PageRank

It obviously did work and they got better results.

It also gave halo of special legitimacy to their results, that they were scientific and unbiased.

# At Microsoft

We gamely expected our version of PageRank to represent about half the overall rank value, largely based on the hype around it.

Turned out it was very expensive to calculate and represented only a small part of the final rank score.

# Agenda

1. Course details.
2. HW1 MostPositiveSubsequence( ).
3. History of the web.
4. Search basics.
5. **Your project.**

# Your project

I've broken it into 8 levels of functionality:

- 0 Basic plan for your project.
- 1 Parse text files into a hash table.
- 2 Build a crawler.
- 3 Build a reverse word index.
- 4 Create a user interface.
- 5 Build a constraint solver and query parser.
- 6 Build a ranker.
- 7 Advanced functionality.

# Next

Overall course plan is:

1. Get you crawling ASAP.
2. Cover the OS topics you'll need.
3. Then, enough to build each piece as you get to it, starting with the crawler.

Next time: Project planning.

Please install GanttProject software.

Who needs a team?