Perspective and course review.

Top 10 scientific algorithms.

Course evaluations.

Final exercises due Tuesday, May 15 at 5pm.
- Individual write-ups.
- Collaboration allowed.

Algorithm. (webster.com)
- A procedure for solving a mathematical problem (as of finding the greatest common divisor) in a finite number of steps that frequently involves repetition of an operation.
- Broadly: a step-by-step procedure for solving a problem or accomplishing some end especially by a computer.

Etymology.
- "algos" = Greek word for pain.
- "algor" = Latin word for to be cold.
- Abu Ja'far al-Khwarizmi's = 9th century Arab scholar.
  - his book "Al-Jabr wa-al-Muqabilah" evolved into today's high school algebra text

A strikingly modern thought.
"As soon as an Analytic Engine exists, it will necessarily guide the future course of the science. Whenever any result is sought by its aid, the question will arise - By what course of calculation can these results be arrived at by the machine in the shortest time?"
Why Does It Matter?

<table>
<thead>
<tr>
<th>Run time (nanoseconds)</th>
<th>1.3 N^3</th>
<th>10 N^2</th>
<th>47 N log_2 N</th>
<th>48 N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to solve a problem of size</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>1.3 seconds</td>
<td>10 msec</td>
<td>0.4 msec</td>
<td>0.048 msec</td>
</tr>
<tr>
<td>10,000</td>
<td>10 seconds</td>
<td>666.67 ms</td>
<td>666.67 ms</td>
<td>2.82 ms</td>
</tr>
<tr>
<td>100,000</td>
<td>1 minute</td>
<td>15 minutes</td>
<td>78 seconds</td>
<td>4.81 seconds</td>
</tr>
<tr>
<td>Million</td>
<td>41 years</td>
<td>2.8 hours</td>
<td>0.94 seconds</td>
<td>48 msec</td>
</tr>
<tr>
<td>10 million</td>
<td>41 millennia</td>
<td>1.7 weeks</td>
<td>11 seconds</td>
<td>0.48 seconds</td>
</tr>
</tbody>
</table>

Max size problem solved in one

<table>
<thead>
<tr>
<th>Seconds</th>
<th>Equivalent</th>
<th>Meters Per Second</th>
<th>Imperial Units</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 second</td>
<td>10^-10</td>
<td>1.2 in / decade</td>
<td>Continental drift</td>
</tr>
<tr>
<td>10</td>
<td>10 seconds</td>
<td>10^-8</td>
<td>1 ft / year</td>
<td>Hair growing</td>
</tr>
<tr>
<td>10^2</td>
<td>1.7 minutes</td>
<td>10^-6</td>
<td>3.4 in / day</td>
<td>Glacier</td>
</tr>
<tr>
<td>10^3</td>
<td>17 minutes</td>
<td>10^-4</td>
<td>1.2 ft / hour</td>
<td>Gastro-intestinal tract</td>
</tr>
<tr>
<td>10^4</td>
<td>2.8 hours</td>
<td>10^-2</td>
<td>2 ft / minute</td>
<td>Ant</td>
</tr>
<tr>
<td>10^5</td>
<td>1.1 days</td>
<td></td>
<td>1</td>
<td>Human walk</td>
</tr>
<tr>
<td>10^6</td>
<td>1.6 weeks</td>
<td></td>
<td>2.2 mi / hour</td>
<td></td>
</tr>
<tr>
<td>10^7</td>
<td>3.8 months</td>
<td></td>
<td>220 mi / hour</td>
<td>Propeller airplane</td>
</tr>
<tr>
<td>10^8</td>
<td>3.1 years</td>
<td></td>
<td>370 mi / min</td>
<td>Space shuttle</td>
</tr>
<tr>
<td>10^9</td>
<td>3.1 decades</td>
<td></td>
<td>620 mi / sec</td>
<td>Earth in galactic orbit</td>
</tr>
<tr>
<td>10^10</td>
<td>3.1 centuries</td>
<td></td>
<td>62,000 mi / sec</td>
<td>1/3 speed of light</td>
</tr>
</tbody>
</table>

N multiplied by 10, time multiplied by

<table>
<thead>
<tr>
<th>Seconds</th>
<th>Equivalent</th>
<th>Meters Per Second</th>
<th>Imperial Units</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 second</td>
<td>10^-10</td>
<td>1.2 in / decade</td>
<td>Continental drift</td>
</tr>
<tr>
<td>10</td>
<td>10 seconds</td>
<td>10^-8</td>
<td>1 ft / year</td>
<td>Hair growing</td>
</tr>
<tr>
<td>10^2</td>
<td>1.7 minutes</td>
<td>10^-6</td>
<td>3.4 in / day</td>
<td>Glacier</td>
</tr>
<tr>
<td>10^3</td>
<td>17 minutes</td>
<td>10^-4</td>
<td>1.2 ft / hour</td>
<td>Gastro-intestinal tract</td>
</tr>
<tr>
<td>10^4</td>
<td>2.8 hours</td>
<td>10^-2</td>
<td>2 ft / minute</td>
<td>Ant</td>
</tr>
<tr>
<td>10^5</td>
<td>1.1 days</td>
<td></td>
<td>1</td>
<td>Human walk</td>
</tr>
<tr>
<td>10^6</td>
<td>1.6 weeks</td>
<td></td>
<td>2.2 mi / hour</td>
<td></td>
</tr>
<tr>
<td>10^7</td>
<td>3.8 months</td>
<td></td>
<td>220 mi / hour</td>
<td>Propeller airplane</td>
</tr>
<tr>
<td>10^8</td>
<td>3.1 years</td>
<td></td>
<td>370 mi / min</td>
<td>Space shuttle</td>
</tr>
<tr>
<td>10^9</td>
<td>3.1 decades</td>
<td></td>
<td>620 mi / sec</td>
<td>Earth in galactic orbit</td>
</tr>
<tr>
<td>10^10</td>
<td>3.1 centuries</td>
<td></td>
<td>62,000 mi / sec</td>
<td>1/3 speed of light</td>
</tr>
</tbody>
</table>

What was COS 423?

Introduction to design and analysis of computer algorithms.
- Algorithmic paradigms.
- Analyze running time of programs.
- Understand fundamental algorithmic problems.
- Intrinsic computational limitations.
- Models of computation.
- Critical thinking.

Material Covered

Algorithmic paradigms.
- Divide-and-conquer.
- Greed.
- Dynamic programming.
- Reduction.

Analysis of algorithms.
- Recurrences and big Oh.
- Amortized analysis.
- Average-case analysis.

Other models of computation.
- On-line algorithms.
- Randomized algorithms.

Intractability.
- Polynomial reductions.
- NP completeness.
- Approximation algorithms.

Fundamental algorithmic problems.
- Sorting and searching.
- Integer arithmetic.
- FFT.
- MST.
- Shortest path.
- Max flow.
- Linear programming.
Top 10 Scientific Algorithms of 20th Century

1. Metropolis Algorithm/ Monte Carlo method (von Neumann, Ulam, Metropolis, 1946). Through the use of random processes, this algorithm offers an efficient way to stumble toward answers to problems that are too complicated to solve exactly.
   - Approximate solutions to numerical problems with too many degrees of freedom.
   - Approximate solutions to combinatorial optimization problems.
   - Generation of random numbers.

Metropolis Algorithm

Local search. Algorithm that explores the space of possible solutions in sequential fashion, moving in one step from a current solution to a "nearby" one.
   - TSP: given a tour, perturb it by exchanging order of two cities.
   - VERTEX-COVER: given a vertex cover, perturb it by adding or deleting a node, so that resulting set remains a cover.

Gradient descent. Replace current solution with neighboring solution that improves objective function, until no such neighbor exists.

- Francis Sullivan

"the greatest influence on the development and practice of science and engineering in the 20th century"

"For me, great algorithms are the poetry of computation. Just like verse, they can be terse, allusive, dense, and even mysterious. But once unlocked, they cast a brilliant new light on some aspect of computing."

Top 10 Scientific Algorithms of 20th Century


"For me, great algorithms are the poetry of computation. Just like verse, they can be terse, allusive, dense, and even mysterious. But once unlocked, they cast a brilliant new light on some aspect of computing."

- Francis Sullivan

"the greatest influence on the development and practice of science and engineering in the 20th century"
Metropolis Algorithm

**Metropolis algorithm.** Gradient descent, but occasionally replace current solution with "uphill" solution.
- Simulate behavior of system according to principles of statistical mechanics.
- Probability of finding a physical system in a state with energy $E$ is proportional to Gibbs-Boltzmann function $e^{-E/(kT)}$, where $T > 0$ is temperature and $k$ is a constant.

**Theorem.** Let $f_S(t)$ be fraction of first $t$ steps in which state of simulation in state $S \in \Sigma$. Then, with probability 1:
\[
\lim_{t \to \infty} f_S(t) = \frac{1}{Z} e^{-E(S)/(kT)},
\]
where $Z = \sum_{S \in \Sigma} e^{-E(S)/(kT)}$.

**Metropolis Step(S)**

- Find neighboring solution $S'$.
- IF $(c(S') \leq c(S))$
  - Update $S \leftarrow S'$.
- ELSE
  - $E \leftarrow c(S') - c(S)$.
  - Update $S \leftarrow S'$ with probability $e^{-E/(kT)}$.

Simulated annealing.
- $T$ large $\implies$ probability of accepting an uphill move is large.
- $T$ small $\implies$ uphill moves are almost never accepted.
- Idea: turn knob to control $T$.
- Cooling schedule: $T = T(i)$ at iteration $i$.

**Physical analog.**
- Take solid and raise it to high temperature, we do not expect it to maintain a nice crystal structure.
- Take a molten solid and freeze it very abruptly, we do not expect to get a perfect crystal either.
- Annealing: cool material gradually from high temperature, allowing it to reach equilibrium at succession of intermediate lower temperatures.

---

Top 10 Scientific Algorithms of 20th Century

   - An elegant solution to a common problem in planning and decision-making: $\max \{cx : Ax \leq b, x \geq 0\}$.
   - One of most successful algorithms of all time.
   - Dominates world of industry.

---

   - A technique for rapidly solving $Ax = b$ where $A$ is a huge $n \times n$ matrix.
   - Conjugate gradient method for symmetric positive definite systems.
   - GMRES, CGSTAB for non-symmetric systems.

**Preconditioned Conjugate Gradient**

```
calculate $x^{(0)} = -Ax^{(0)}$ for some initial guess $x^{(0)}$
for $i = 1, 2, ...$
    calculate $Mx^{(i-1)} = x^{(i-1)}$
    $\mu_{i-1} = x^{(i-1)^T} \mu_{i-1}$
    $\sigma_{i-1} = \sigma_{i-1} + 1$
    $x^{(i)} = x^{(i-1)}$
    else
        $\sigma_{i-1} = \sigma_{i-1}/\mu_{i-1}$
        $\mu_{i-1} = \mu_{i-1} - \mu_{i-1} \sigma_{i-1}$
        $\sigma_{i-1} = \sigma_{i-1} - \sigma_{i-1}$
        end
    if $\|Ax - b\| < \epsilon$
        break
    else
        $\mu_{i-1} = \mu_{i-1}/\sigma_{i-1}$
        $\sigma_{i-1} = \sigma_{i-1} + 1$
        $x^{(i)} = x^{(i-1)}$
    end
end
```
Top 10 Scientific Algorithms of 20th Century

   - Factor matrices into triangular, diagonal, orthogonal, tri-diagonal, and other forms.
   - Analysis of rounding errors.
   - Applications to least squares, eigenvalues, solving systems of linear equations.
   - LINPACK, EISPACK.

5. Fortran Optimizing Compiler (Backus, 1957). Turns high-level code into efficient computer-readable code.
   - Among single most important events in history of computing: scientists could program computer without learning assembly.

```fortran
500   C = 0.0
C  *** START LOOP ***
DO 540 I=L,K
   F = S*RV1(I)
   RV1(I) = C*RV1(I)
   IF (ABS(F).LE.EPS) GO TO 550
   G = W(I)
   H = SQRT(F*F+G*G)
   W(I) = H
   C = G/H
   S = -F/H
510   CONTINUE
```

6. QR Algorithm for Computing Eigenvalues (Francis 1959). Another crucial matrix operation made swift and practical.
   - Eigenvalues are arguably most important numbers associated with matrices.
   - Differential equations, population growth, building bridges, quantum mechanics, Markov chains, web search, graph theory.

\[
A x = \lambda x
\]

**QR(A)**

\[
A_{k+1} = R_k Q_k
= Q_k^{-1} Q_k R_k Q_k
= Q_k^{-1} A_k Q_k
\]

⇒ \(A_{k+1}\) and \(A_k\) have same eigenvalues

Under fairly general conditions, \(A_k\) converges to diagonal or upper triangular matrix with eigenvalues on main diagonal.
Web Search

Some "authoritative" pages (obtained from Kleinberg algorithm):

- [www.eff.org](www.eff.org) (Electronic Frontier Foundation)
- [www.cdt.org](www.cdt.org) (Center for Democracy and Technology)
- [www.vtw.org](www.vtw.org) (Voters Telecommunications Watch)
- [www.aclu.org](www.aclu.org) (American Civil Liberties Union)

Authoritative page: need quantitative definition.
- Non-trivial problem: query for "search engine" unlikely to report Yahoo, Excite, or AltaVista since they do not use the term.
- Yahoo solution: legion of human catalogers.
  - page p points to q: creator of page p confers authority on q
  - pitfalls: navigational links, relevance vs. popularity

Hubs and Authorities

Good hub: page that points to many good authorities.
Good authority: page pointed to by many good hubs.

Iterative algorithm: authority weights \( x(p) \), and hub weights \( y(p) \).
- Set authority weights \( x(p) = 1 \), and hub weights \( y(p) = 1 \) for all \( p \).
- Repeat following two operations (and then re-normalize \( x \) and \( y \) to have unit norm):

\[
\begin{align*}
    x(p) &= \sum_{q: q \text{ points to } p} y(q) \\
    y(q) &= \sum_{p: p \text{ points to } q} x(p)
\end{align*}
\]

Hubs and Authorities

Theorem (Kleinberg, 1997). The iterates \( x(p) \) and \( y(p) \) converge to the principal eigenvectors of \( A^T A \) and \( A A^T \), where \( A \) is the adjacency matrix of the (directed) Web subgraph.
- Algorithm is essentially "Power method" for computing principal eigenvector.
- Can use any eigenvector algorithm, e.g., QR algorithm.

Web Search: Clustering

Principal eigenvector.
- [www2.ecst.csuhcico.edu/~jaguar.html](www2.ecst.csuhcico.edu/~jaguar.html) (404 Not Found)
- [www.mcc.ac.uk/dlms/~du/~jaguar.html](www.mcc.ac.uk/dlms/~du/~jaguar.html) (Jaguar Page)

2nd non-principal eigenvector: positive components.
- [www.jaguarsnfl.com](www.jaguarsnfl.com) (Jacksonville Jaguars NFL)
- [www.nando.net/~jax.htm](www.nando.net/~jax.htm) (Jacksonville Jaguars Home Page)

3rd non-principal eigenvector: positive components.
- [www.jaguarvehicles.com](www.jaguarvehicles.com) (Jaguar Cars Global Home Page)
- [www.collection.co.uk](www.collection.co.uk) (The Jaguar Collection)
Web Search: Clustering

2nd non-principal eigenvector: positive components.
- [www.caral.org/abortion.html](http://www.caral.org/abortion.html) (Abortion and Reproductive Rights)
- [www.plannedparenthood.org](http://www.plannedparenthood.org) (Welcome to Planned Parenthood)
- [www.gynpages.com](http://www.gynpages.com) (Abortion Clinics Online)
- [www.prochoice.org/naf](http://www.prochoice.org/naf) (National Abortion Federation)

2nd non-principal eigenvector: negative components.
- [www.awinc.com/.../lifenet.htm](http://www.awinc.com/.../lifenet.htm) (LifeWEB)
- [www.worldvillage.com/.../peter.htm](http://www.worldvillage.com/.../peter.htm) (Healing After Abortion)
- [www.members.aol.com/pladvocate](http://www.members.aol.com/pladvocate) (Pro-Life Advocate)
- [www.catholic.net/.../abortion.html](http://www.catholic.net/.../abortion.html)

Top 10 Scientific Algorithms of 20th Century

7. Quicksort (Hoare, 1962). Given N items over a totally order universe, rearrange them in increasing order.
   - O(N log N) instead of O(N^2).
   - Efficient handling of large databases.

8. Fast Fourier Transform (Cooley, Tukey 1965). Perhaps the most ubiquitous algorithm in use today, it breaks down waveforms (like sound) into periodic components.
   - O(N log N) instead of O(N^2).

9. Integer Relation Detection (Ferguson, Forcade, 1977). Given real numbers x_1, ..., x_n, find integers a_1, ..., a_n (not all 0 if they exist) such that a_1x_1 + ... + a_nx_n = 0?
   - PSLQ algorithm generalizes Euclid’s algorithm: special case when n = 2.
   - Find coefficients of polynomial satisfied by 3rd and 4th bifurcation points of logistic map.
   - Simplify Feynman diagram calculations in quantum field theory.
   - Compute n^{th} bit of \pi without computing previous bits.
   - Experimental mathematics.

10. Fast Multipole Method (Greengard, Rokhlin, 1987). Accurate calculations of the motions of N particles interacting via gravitational or electrostatic forces.
   - Central problem in computational physics.
   - O(N) instead of O(N^2).
   - Celestial mechanics, protein folding, etc.
11. Newton’s method (Newton, 16xx). Given a differentiable function $f(x)$, find a value $x^*$ such that $f(x^*) = 0$.
   - Start with initial guess $x_0$.
   - Compute a sequence of approximations: $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$.
   - Equivalent to finding line of tangent to curve $y = f(x)$ at $x_i$ and taking $x_{i+1}$ to be point where line crosses x-axis.

Kevin’s Non-Scientific Honorable Mention

12. Depth first search (Tarjan). Learn properties of a graph by systematically examining each of its vertices and edges.
   - Connectivity.
   - Cycle detection.
   - Bipartiteness.
   - 2-SAT, 2-colorability.
   - Topological sort.
   - Transitive closure.
   - Euler tour.
   - Bi-connectivity.
   - Strong connectivity.
   - Planarity.

   - Two different keys: Alice’s PUBLIC key locks, her PRIVATE key opens. Everything else is public.
RSA Public-Key Cryptosystem

Key generation.
- Select two large prime numbers p and q at random.
- Compute \( n = pq \), and \( \phi = (p-1)(q-1) \).
- Choose integer e that is relatively prime to \( \phi \).
- Compute \( d \) such that \( de \equiv 1 \pmod{\phi} \).
- Publish \((e, n)\) as public key.
- Keep \((d, n)\) as secret key.

\[
\begin{align*}
p &= 11, \\
q &= 29 \\
n &= 319, \\
\phi &= 280 \\
e &= 3, \\
d &= 187 \\
M &= 100
\end{align*}
\]

RSA Example

Parameters.
- \( p = 47, \\
q = 79, \\
n = 3713, \\
\phi = 3588 \\
e = 17, \\
d = 3377 \\
M = 2003 \]

Modular exponentiation.
- \( 2003^{17} \pmod{3713} = 2003^{16} * 2003^1 \pmod{3713} = 2003^{16} \pmod{3713} \)
- \( 2003^{8} \pmod{3713} = 2003^4 \pmod{3713} = 2003^2 \pmod{3713} = 2003 \pmod{3713} \)
- \( 2003^{16} \pmod{3713} = 3157 \)

Efficient alternative (repeated squaring).
- \( 2003^1 \pmod{3713} = 2003 \)
- \( 2003^2 \pmod{3713} = 4012009 \pmod{3713} \pmod{3713} = 1969 \)
- \( 2003^3 \pmod{3713} = 1969^2 \pmod{3713} = 589 \)
- \( 2003^6 \pmod{3713} = 589^2 \pmod{3713} = 1612 \)
- \( 2003^{16} \pmod{3713} = 3157 \)

RSA Public-Key Cryptosystem

Bob sends message M to Alice.
- Bob obtains Alice’s public key \((e, n)\) from Internet.
- Bob computes \( C = M^e \pmod{n} \).

Alice receives message C.
- Alice uses her secret key \((d, n)\).
- Alice computes \( M' = C^d \pmod{n} \).

Why does it work? Need \( M = M' \). Intuitively.
- \( M = M'^d \pmod{n} = M^{ed} \pmod{n} = M \) Recall: \( ed \equiv 1 \pmod{\phi} \).
- Argument not rigorous because of mod.
  - rigorous argument uses fact that p and q are prime and \( \phi = (p-1)(q-1) \)

RSA Details

How large should \( n = pq \) be?
- 1024 bits for long term security.
- Too small \( \Rightarrow \) easy to break.
- Too large \( \Rightarrow \) time consuming to encrypt/decrypt.

How to choose large "random" prime numbers?
- Miller-Rabin procedure checks whether \( x \) is prime. Usually!
  - Guess, and use subroutine to check.
- Number theory \( \Rightarrow n / \log n \) prime numbers between 2 and \( n \).
  - Primes are plentiful: \( 4.3 \times 10^97 \) with \( \leq 100 \) digits.

How to compute \( d \) efficiently?
- Existence guaranteed since \( \gcd(e, \phi) = 1 \).
- Fancy version of Euclid’s algorithm.
Extra Slides

RSA Public-Key Cryptosystem

Why does it work? Rigorously.

- \( M' = C^d \pmod{n} \)
- \( = M^{ed} \pmod{n} \)

Now, since \( \phi = (p-1)(q-1) \) and \( e \) \( d \equiv 1 \pmod{\phi} \)

\( ed = 1 + k(p-1)(q-1) \) for some integer \( k \).

A little manipulation.

- \( M^{ed} \equiv M M^{(p-1)(q-1)} \pmod{p} \)
- \( = M (1)^{k(q-1)} \pmod{p} \)
- \( = M \pmod{p} \)

(trivially true if \( M \equiv 0 \))

- \( M^{ed} \equiv M \pmod{q} \)

Finally.

- \( M^{ed} \equiv M \pmod{pq} \)

Fermat's Little Theorem

if \( p \) is prime, then for all \( a \neq 0 \)

\( a^{p-1} \equiv 1 \pmod{p} \)

Chinese Remainder Theorem

if \( p, q \) prime then for all \( x, a \)

\( x \equiv a \pmod{pq} \iff x \equiv a \pmod{p}, x \equiv a \pmod{q} \)