Computational Thinking

IS 101Y/CMSC 104Y
First Year IT

Penny Rheingans
University of Maryland Baltimore County

What are important problems?

- **Break into your groups**
- **Each group should come up with ten problems/goals in world/life**
  - **Could be important societal challenges**
  - **Could be something important to you**
  - **Could be something you enjoy**
- **Report out by random group member**

Important problems

Important problems

Important problems

Toolsmith

Toolsmith

- What were main points of this article?
- Do you agree?
- This article is 16 yrs old. Are there some things that have changed?

Computational Thinking

Computational Thinking

- What were main points of this article?
- Do you agree?
- Does this view support or contradict that of Brooks?
- Should everyone learn about CT? When?

CS Principles: Big Ideas

- Anybody take AP CS?
  - Why/why not?
- Joint effort between CRA and College Board to develop new AP CS Principles course
  - The computing content of this course generally follow that model
- Big Ideas
  - Creativity
  - Abstraction
  - Data
  - Algorithms
  - Programming
  - Internet
  - Impact

Computing is a <u>creative</u> activity.

- Computing fosters the creation of artifacts.
  - Computing enables people to create digitally—including creating knowledge, tools, expressions of ideas, and solutions to problems.
  - Computing enables people to translate intention into digital artifacts.
- Computing fosters creative expression.
  - Computing extends traditional forms of human expression and experience.
  - Computing fosters the creation of new forms of expression.
  - Computing enables creative exploration that informs and inspires.
- Programming is a creative process.
  - Some programs are developed to satisfy personal curiosity or for creative expression.
  - Some programs are developed to solve problems, develop new knowledge, or help people, organizations, or society.

<u>Abstraction</u> reduces information to focus on relevant concepts.

- A combination of abstractions built upon binary sequences can be used to represent all digital data.
  - interpretation depends on how it is used (instr, #, txt, snd, or img).
  - finite representation used for infinite mathematical concept of a number.
  - Number bases are abstractions used for reasoning about digital data.
- Multiple levels of abstraction are used in computation.
  - layers of computing hardware, including gates, chips, and components.
  - Programming languages, low to high level, used in developing software.
  - Systems use levels of hardware, software, and conceptual abstractions.
- Models and simulations use abstraction to raise and answer questions.
  - People use models to generate new understanding and knowledge.
  - Models use different levels of abstraction to represent phenomena.
  - Hypotheses can be formulated, refined, and tested using models.
  - Simulations can facilitate extensive and rapid testing of models.

Data and information facilitate the creation of knowledge.

- People use computer programs to process information to gain insight and knowledge.
  - find patterns in, and test hypotheses about, digital information.
  - translating and transforming digitally represented information.
- Computing facilitates exploration and the discovery of connections in information.
  - *Big Data* provides new opportunities and new challenges.
  - Scalability, of systems and approach, is key when datasets are large.
  - Metadata can provide more information about other aspects of the data.
- Computational manipulation of information requires consideration of representation, storage, security, and transmission.
  - trade-offs in digital representation of digital and non-digital info.
  - many formats depending on its size, intended use, etc.

Algorithms are used to express solutions to computational problems.

- An algorithm is a precise sequence of instructions for a process that can be executed by a computer.
  - Sequencing, selection, iteration, and recursion are building blocks.
  - Different algorithms can be developed to solve the same problem.
- Algorithms are expressed and implemented using languages.
  - natural language, pseudo-code, and visual and textual languages.
  - better suited for expressing different algorithms.
  - can affect clarity or readability, but not whether solution exists.
- Algorithms can solve many, but not all, problems.
  - Many problems can be solved in a reasonable time.
  - Some need heuristic approaches to solve them in a reasonable time.
  - Some problems cannot be solved using any algorithm.
- Algorithms are evaluated analytically and empirically.
  - using many criteria (e.g., efficiency, correctness, and clarity).
  - algorithms for the same problem can have different efficiencies.

Programming enables problem solving, expression, and knowledge creation.

- Programs are written to execute algorithms.
  - Requires an understanding of how instructions are processed.
  - Programs are executed to automate processes.
  - A single program can be run multiple times and on many machines.
  - Executable programs increase the scale of problems that can be solved.
- Programming is facilitated by appropriate abstractions.
  - Functions are re-usable programming abstractions.
  - Parameterization can be used to generalize a specific solution.
  - Data abstraction can separate behavior from implementation.
  - APIs and libraries simplify complex programming tasks.
- Programs are developed and used by people.
  - Developing programs is an iterative process.
  - Finding and eliminating errors is an essential part.
  - Documentation is necessary for developing maintainable programs.
  - Programs are evaluated for their correctness and style.

The Internet pervades modern computing.

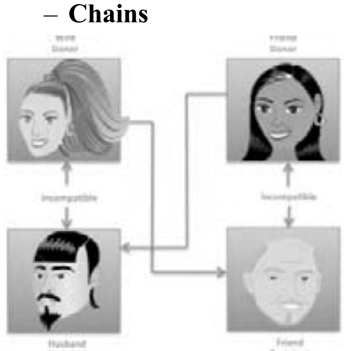
- The Internet is a network of autonomous systems.
  - The Internet connects devices and networks all over the world.
  - The Internet and the systems built on it facilitate collaboration.
  - The Internet is built on evolving standards for addresses and names.
- Characteristics of the Internet and the systems built on it influence their use.
  - Hierarchy and redundancy help systems scale.
  - Interfaces and protocols enable widespread use.
  - The size and speed of systems affect their use.
- Cybersecurity is an important concern for the Internet and systems built on it.
  - The trust model of the Internet involves tradeoffs.
  - Cryptography is essential to many models of cybersecurity.
  - Has software, hardware, and human components.

Computing has global impacts.

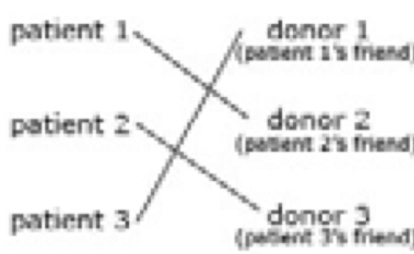
- Computing affects communication, interaction, and cognition.
  - enhances communication; new ways to communicate and collaborate.
  - facilitates id of problems, creation of solutions, and telling of results.
  - enhances human capabilities (with cyber-physical sys and assistive tech).
  - The Internet and the web have a profound impact on society.
- Computing enables innovation in nearly every field.
  - Computational approaches and data analysis enable innovation.
  - enables innovation by providing access to, and sharing of, information.
- Computing has both beneficial and harmful effects.
  - Innovations enabled by computing raise legal and ethical concerns.
  - Privacy and security concerns arise in the development and use of tech.
  - collection, use, and exploitation of info about persons, groups, and inst.
  - raises questions about intellectual property.
- Computing exists within economic, social, and cultural contexts.
  - innovations both influence and are influenced by their contexts.
  - issues of equity, access, and power.

A Kidney Story

- **Kidney disease affects 50,000 new Americans a year**
- **Transplants as treatment**
  - **Pairs**
  - **Cycles**
  - **Chains**



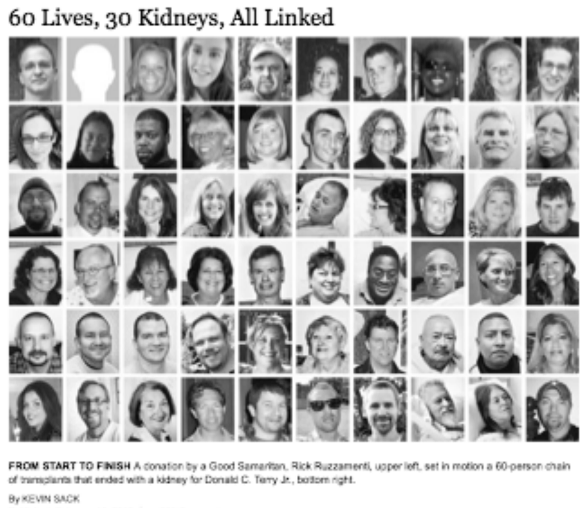
Two cycle



Three cycle

A Big Kidney Story

What about really big chains?



How do you come up with optimal series of swaps?

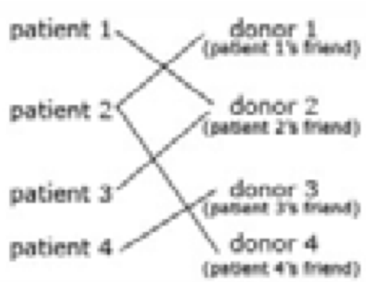
NY Times, Feb 9, 2012

For more info:

– **http://www.nsf.gov/cise/csbytes**

Kidney Exchange

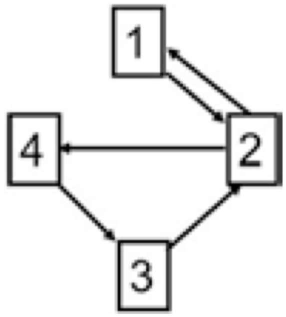
- **Consider the exchange below. A patient is connected to a donor if they are biologically compatible. A donor will only donate a kidney if his or her friend also receives a kidney. What is the optimal matching for this exchange? Why?**



- **What technique did you use to solve this problem? How would your technique scale if there were ten donors and patients? 100? Thousands?**

Alternative Representation

- A graph data structure can capture the important relationships among patients and donors.



- A legal exchange is one where there is a path following edges that visits each vertex exactly once and returns to the starting vertex (if every node = Hamiltonian cycle, http://nrich.maths.org/2320).
- What is the longest such path in this graph?