Internet

domain — region of Internet operated by 1 entity (university, company, etc.)
domain name — assigned by registrars

Top-level domains — .edu, .com, .dk
Example: login.imada.sdu.dk — imada is a subdomain

IP addresses:

- IPv4: 32 bits: 10.110.4.199
- IPv6: 128 bits: 2001:0DB8:AC10:FE01 — hexadecimal (first half shown)

Domain name server (DNS) — Internet directory

212.97.129.250 vs. www.sdu.dk
IP addresses

IP addresses: IPv4: 32 bits: 10.110.4.199

Which number base are IPv4 addresses written in?
How large can a number between dots be?

A. decimal, less than 256 between dots
B. hexadecimal, less than 256 between dots
C. decimal, less than 512 between dots
D. hexadecimal, less than 512 between dots
E. decimal, less than 1024 between dots

Vote at m.socrative.com. Room number 415439.
IP addresses: IPv4: 32 bits: 10.110.4.199

Which number base are IPv4 addresses written in?
How large can a number between dots be?

A. decimal, less than 256 between dots
Application: email

Some protocols involved:

▶ SMTP — sending e-mail between machines
▶ MIME — make data compatible with SMTP
▶ accessing e-mail
  ▶ POP3 — mail transferred to your own computer
  ▶ IMAP — mail stays on mail server
    — can access mail from other computers

Try looking at full header for some email. How many intermediate machines did it go through?
Protocols

Layered models
— abstraction to handle complexity

Communication protocols at layer $N$
— see virtual machine connection at layer $N-1$.
— invoke facilities at layer $N-1$ to transmit layer $N$ data units.
Protocols

ISO Open System Interconnection Model (OSI)
vs.
Internet Model — TCP/IP
Protocols

Internet Model — TCP/IP

- **Application** — ssh, sftp, HTTP, SMTP
- **Transport** — converts messages to packets, orders packets
  - **TCP** — transmission control protocol
    - establishes a connection before sending
    - messages and acknowledgements
    - example: e-mail
  - **UDP** — user datagram protocol
    - no connection established — example: VoIP
- **Network** — IP — internet protocol
  - converts packets to datagrams
  - assigns intermediate addresses
- **Link** — transfers packets
Protocols

Internet Model — TCP/IP

Messages sent through a path in Internet.

Going from one machine to the next — hop

In intermediate stops for a message, only lower layers involved.

Determining which application protocol should get incoming message
— port number — 80 is HTTP
Hands-on Internet

Start a command prompt.

(Win 8: Win-X, choose command prompt, Win 7: Search for “command” in start button, Ubuntu (Unity): search for “terminal” in Ubuntu-button (top, left), Mac OS X: search (top, right) for “terminal”).

Try the following commands:

- Show network interface info: ipconfig /all; ifconfig; /sbin/ifconfig
- Show active connections: netstat
- Contact host: ping google.com
- Show route to host: tracert google.com; traceroute google.com

(Some must be stopped by “CNTL C”)
Browsers

World Wide Web (WWW) — for making information available.

Which browser do you use most?

A. Firefox
B. Internet Explorer
C. Chrome
D. Opera
E. Safari

Vote at m.socrative.com. Room number 415439.

No correct answer.
hypertext — text documents containing hyperlinks.

hypermedia — more than text (audio and/or video)

Hypertext Transfer Protocol (HTTP)
— to get Web pages displayed by your browser

HTTPS — using SSL or TLS — Transport Layer Security

URL = Uniform Resource Locator — address

Example: http://imada.sdu.dk/~joan/intro/15slides5.pdf

protocol://host with document/directory path/file (document)

HTML — Hypertext Markup Language — can include JPEG, etc.

XML — more general than text
— standardized style organizing and making searching easy
— for recipes, one markup language — for music another
Different systems for server-side or client-side functionality.

PHP, ASP, JSP for server side functionality
(database operation, for example)

JavaScript, Applets, Flash — to run programs on client side

Security problem — running programs from elsewhere
Algorithm: a well-ordered collection of unambiguous and effectively computable operations, that, when executed, produces a result in a finite amount of time.
Algorithm: a well-ordered collection of unambiguous and effectively computable operations, that, when executed, produces a result in a finite amount of time.

Examples:

- computing with floating point numbers
- compressing data
- executing machine code
Algorithms

**Algorithm**: a well-ordered collection of unambiguous and effectively computable operations, that, when executed, produces a result in a finite amount of time.

**Examples**:
- computing with floating point numbers
- compressing data
- executing machine code

**Program**: representation of an algorithm

**Pseudocode**: representation of an algorithm

**Process**: execution of an algorithm
Art of problem solving

Polya’s principles applied to algorithms:

1. Understand the problem
2. Get an idea for a possible algorithmic procedure (to solve it)
3. Formulate the algorithm and represent it as a program
4. Evaluate the program for correctness and its potential as a tool for solving other problems
Art of problem solving

Polya’s principles applied to algorithms:

1. Understand the problem
2. Get an idea for a possible algorithmic procedure (to solve it)
3. Formulate the algorithm and represent it as a program
4. Evaluate the program for correctness and its potential as a tool for solving other problems

Not so easy as 1 $\rightarrow$ 2 $\rightarrow$ 3 $\rightarrow$ 4.
Examples:

- Magic trick — ideas, discover they don’t work with some initial cards...
- 3 politicians (no names) A, B, C — know each other
  - 1 always tells the truth
  - 1 always lies
  - 1 does some of each
  - Ask 3 true/false questions
    - choose whichever politician you like for whichever question
    - determine which politician is which
Algorithm design techniques

Techniques:

- Brute force
- Stepwise refinement (top-down)
  - break into smaller and smaller problems
  - if modular (relatively independent) parts, can program in teams — software engineering
Cute problems in textbook.

Example: Step from pier into a boat
Hat falls into water.
River flows 2.5 miles/hour
Go upstream at 4.75 miles/hour
After 10 minutes discover hat missing.
Turn around to travel downstream.
How long before you get to the hat?
Cute problems in textbook.

Example: Step from pier into a boat
Hat falls into water.
River flows 2.5 miles/hour
Go upstream at 4.75 miles/hour
After 10 minutes discover hat missing.
Turn around to travel downstream.
How long before you get to the hat?
Answer: 10 minutes
— It pays to think.
Pseudocode

- easier to read than a program
- syntax less important
- constructs from many languages work the same
Pseudocode

- easier to read than a program
- syntax less important
- constructs from many languages work the same
  - if...then...else — condition is Boolean
  - while
  - repeat
  - for
  - recursion
Types — use consistently and clearly

Incorrect example: Card := Card + n
Pseudocode

Types — use consistently and clearly

Incorrect example: Card := Card + n
Incorrect example: Suppose Card has the form \((s_1, v_1)\) and \(1 \leq n \leq 6\).

Must explain the general idea and what variables are used for if not obvious — not what it does, but why, in if...then...else clause for example.
Sequential search

**Sequential search problem:**

**Input:** List of elements, TargetValue

**Output:** success if TargetValue is in List

failure if it is not in List

A brute force algorithm.
Sequential search

**procedure** `Search(List, TargetValue)`:  
{ Input: List is a list; TargetValue is a possible entry }  
{ Output: *success* if TargetValue in List; *failure* otherwise }

```prolog
if (List empty)
  then Output failure

else
  TestEntry := 1st entry in List
  while (TargetValue ≠ TestEntry and there are entries not considered)
    (TestEntry := next entry in List)
  if (TargetValue = TestEntry)
    then Output success
  else Output failure
```
Sequential search

Analysis:

- time
- fundamental operation
  - takes time
  - number of occurrences proportional to everything else that happens
Sequential search

Analysis:

| List       | =  n |

How many comparisons are necessary in the worst case?

A. 1
B. $n - 1$
C. $n$
D. $n + 1$
E. $2n$

Vote at m.socrative.com. Room number 415439.
Analysis:

| List | =  \( n \)

How many comparisons are necessary in the worst case?

D.  \( n + 1 \)

This is \( \Theta(n) \).