

**Source:**



## 1 | **rand ideas in other sciences**

1.1 | **theory of matter**

1.2 | **big bang theory**

1.3 | **newtons laws**

1.4 | **conservation of matter / energy**

1.5 | **cell theory**

1.6 | **evolution**

1.7 | **math**

1.7.1 | **finding relationships (abstract things)**

## 2 | **what do those grand theories do?**

2.1 | **describe invariant relationships like  $E=mc^2$**

2.2 | **define limits on what is possible and what isn't**

2.3 | **emergent properties from computational systems that are difficult to predict**

## 3 | **how does computing let us do similar things to laws and theories in science?**

## 4 | **computational complexity theory**

4.1 | **how long it takes to compute the answer as a function of the input size**

4.2 | **overview of presentation**

4.2.1 | **methods for determining computational complexity**

4.2.2 | **wide variation in complexity of diff problems**

4.2.3 | **computationally hard problems are very difficult**

4.2.4 | **some problems have not yet been proven**

4.2.5 | **problems have been grouped into equivalence classes**

## 5 | **big O notation**

5.1 | **approximate run time (not exact)**

5.2 | **how the time scales/changes**

2. structure and interpretation of computer programs source recommended



**7 | programs are complex (more words than war and peace)**

**8 | programming can become faster by developing tools**

**8.1 | languages, compilers, debuggers, editors, libraries, methodologies, code repos**

**9 | missing grand idea: evaluating languages scientifically**

**9.1 | people adopt languages in a bandwagon-ey way**

**9.2 | people compared lisp and java and found that lisp tended to be faster, faster to write, and shorter**

**10 | inspire human reasoning skills from computation**

**10.1 | computational thinking by jeannette wing**

**11 | the internet – communications network that interconnects almost every computer on earth**

**11.1 | design goals**

**11.1.1 | highspeed**

**11.1.2 | reliable / decentralized**

**11.1.3 | many types of computers**

**11.1.4 | many types of networking tech**

**11.1.5 | no application knowledge of network tech**

**11.1.6 | no application knowledge of networking topology**

**11.1.7 | many applications**

**11.1.8 | simple application interface**

**11.1.9 | anonymity**

**11.1.10 | security**

**11.2 | design solution**

**11.2.1 | packet switching over circuit switching (wires don't move)**

**11.2.2 | some redundancy**

**11.2.3 | common packet protocol**

**11.2.4 | routing algorithms**