Suggestive Topics: Observation • Software Implementation • 'Language of Mathematics' • Abstract Data Types (arrays, LLs, queues/stacks, BSTs, hash tables) • Sorting and Selection • Asymptotic Notations • Philosophical Paradigms • Models of Computation • Optional Assignment • "Time Management"

Minds Matter Philadelphia, Math Session #4 (2021-22)

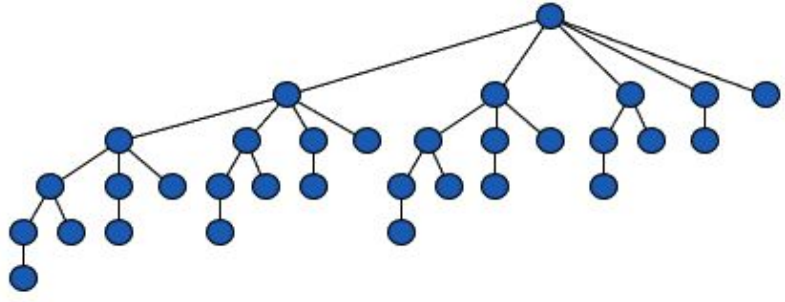
Algorithmic Problem-Solving [recording]

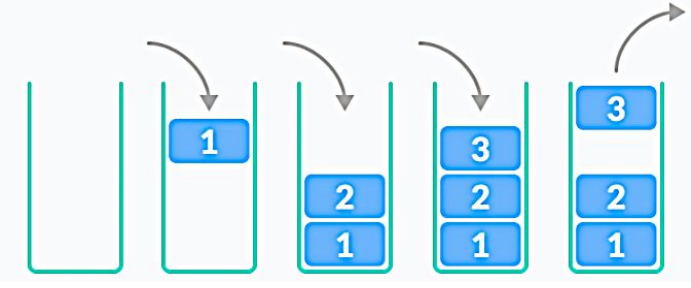
Saturday Math Tutoring Activities

Dachao⁺ (presenter)

Kathy (co-host)

$$(\pi < \pi^2) \Rightarrow (\pi = 5)\ ?$$





(Programiz, https://www.programiz.com/dsa/stack)

Minds Matter Philadelphia Sophomores

with Class of 2024, Fall 2021

December 4, 2021

10:45-11:30 a.m.

⁺Correspondence to Ms. Nancy at nancy.olewnik@mindsmatterphilly.org

About Me (Dachao Sun)

Current Graduate Student/Inquirer in Literacy, Reading/Writing Education at the University of Pennsylvania (2021-22)

Computing Science: Computer Graphics, and Image Analysis and Computational Photography Algorithms (2014-18)

Engineering Student in College (pre-2014)

Penn GSE
GRADUATE SCHOOL OF EDUCATION
UNIVERSITY of PENNSYLVANIA

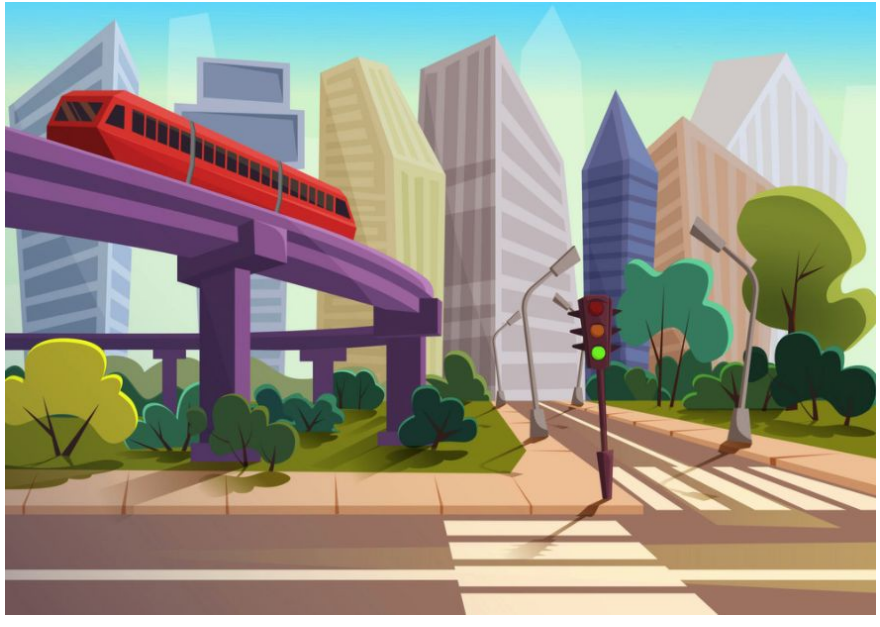
HULT
INTERNATIONAL
BUSINESS SCHOOL

CLEMSON

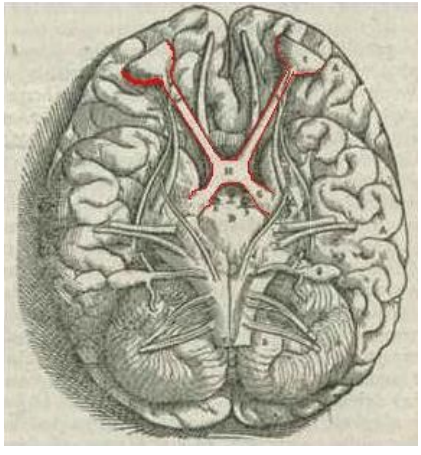
Observation

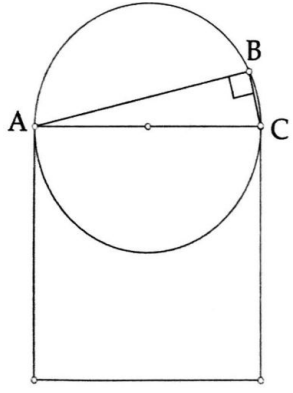
Ob`serve´: It means to take **notice** of by appropriate conduct; to conform one's action or practice to; to keep; to *heed* (i.e. mind, regard with care) (Webster's 1913 Dictionary)

E.g. The number of footsteps it takes from bedroom to answer a door bell.



(VectorStock, https://www.vectorstock.com/royalty-free-vector/cartoon-modern-city-panorama-with-glass-vector-21504138)





Optical Chasm

(Wikipedia, image from Andreas Vesalius' Fabrica, 1543)

Smallest among {70, 31, 9, 78, 36, 73, 50, 40, 21, 75} ?

Algorithms

(From Wikipedia) The word *algorithm* is derived from the name of the 9th-century Persian mathematician Muḥammad ibn Mūsā al-Khwārizmī, whose *nisba*, an adjective indicating the person's place of origin etc. in Arabic names (identifying him as from Khwarazm) was Latinized as Algoritmi.

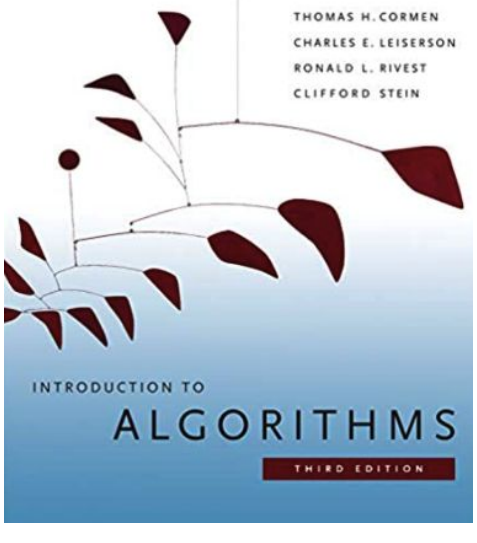
An algorithm is a precise rule (or set of rules) specifying **how to solve some problem**; a set of **procedures *guaranteed*** to find the solution to a problem.

(Webster's 1913 Dictionary)

A recipe (a conceptual one) to do some task.

Essential qualifications for a “good” algorithm:

- Always terminates, and produces **correct** results
- **Efficient**: leverage of computational resources (minimizes running time, memory, network bandwidth..)
- **Simple** to describe, “paraphrase”, and analyze



Data Structures / Two Basic Abstract Data Types

Data structures are the actual **containers of data elements**, a little bit similar to the role of symbolic placeholders in algebra/arithmetic.

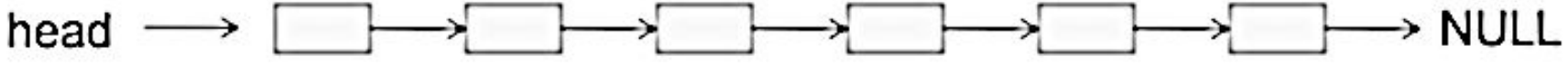
Sets/Dictionaries−an “abstract” data type: one that supports the following operations,

1. Insert; 2. Remove; 3. Find

Arrays: allows retrieval or modification of any element in “O(1)” time.

Linked Lists: related questions are in particular popular in job interviews, e.g.

- Insert or delete a node in a linked list
- Reverse a linked list



C++ Programming Syntaxes

main.cpp

```cpp
#include <iostream> // premable: some "libraries" included
#include <stdlib.h>
//using namespace std;

int main()
{
    // Declare a variable & specify its value
    // using the "assginment" ("=") operator
    int b = 37;

    // "Print" stuff to console
    std::cout << b << "\n";

    // Explicit declaration of an array (of five integers)
    int array[5] = {2, 0, 2, 1, 999};
}
```

```
37

...Program finished with exit code 0
Press ENTER to exit console.
```

Semi-colons; equality sign as assignment;

"Character OUTput";

"//" or "/* */" to comment;

Brackets "[.]" can be thought of as a symbolic notation for **indexing**;

Parentheses suggest to call/use an already written modular routine of code, a.k.a. *function*

There is a "terminal" or "console" for results

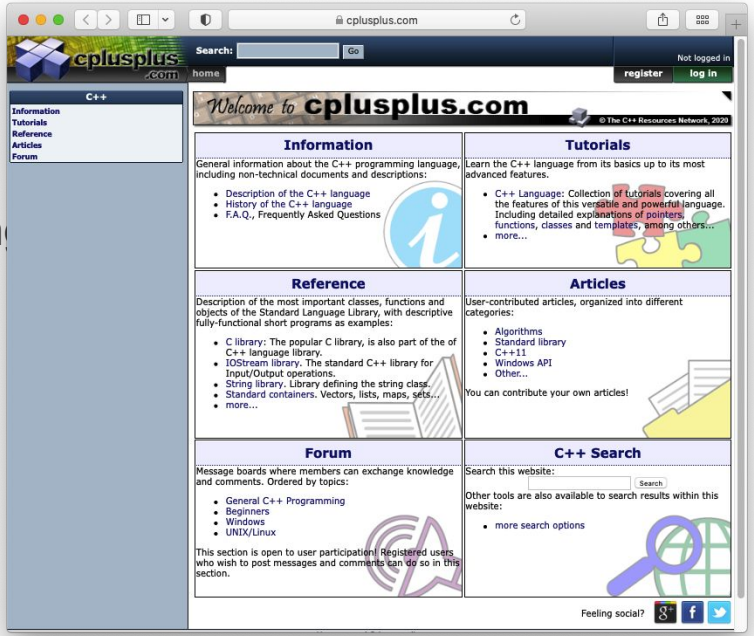
Divergence: Implementing a “Data” Problem on Computer

Realization or *implementation* (represent by instance) of a problem could be fun.

We need some data first. How to generate a dozen of (pseudo-)random integers?

Craft such data idea in programming language C++ (1985-).

- GDB Online (primary): http://www.onlinegdb.com
- OneCompiler: http://onecompiler.com/cpp
- Programiz: http://www.programiz.com/cpp-programmin



Exercise: Try to **make an array** for the following

{83, 86, 77, 15, 93, 35, 86, 92, 49, 21, 62, 27,
90, 59, 63}, i.e. set of 15 integers

And, calculate their **average** and **print it out**.

Sorting and Selection (by rank)

"Have you got everything sorted?"

Thinking about the process of sorting, we probably need to (know how to) **compare** elements of objects, e.g. two integers.

Sorting (a set of objects or a "sequence", not only numberable but also *comparable*) == to shuffle them around until somehow things are all in ascending (or descending) order in that each element is "smaller" (or "larger") than the next consecutive one.

Insertion Sort -- Scan the sequence and find-and-put the smallest element at front.

Bubble Sort -- Keep comparing consecutive pairs of elements, swapping the larger one "up" into a smaller-larger pair, all the way up till the end of sequence, and repeat.

Merge Sort -- Break sequence into sub-parts, "recursively" sort each, and merge them together in a way such that the outcome is also sorted (divide and conquer).

QuickSort -- Partition the sequence keyed on a "pivot" element, recursively sort halves

Selection == given a specified "rank" (the position in the ordered sequence), retrieve the corresponding element of data.

Example: Pseudocode Description of "Bubble Sort"

The *pseudocode* of an algorithm is a "plain language" description of the steps in it or another hardware/software system that implements it.

An algorithm of sorting, then, applies on a (generically speaking) sequence by **changing (via shuffling or re-positioning) the orders of elements until they are sorted**, either ascending or descending, such that any **consecutive pairs of elements are all in the increasing** (or all decreasing) **order**.

Question: Is this sequence sorted? {3, 1, 4, 1, 5, 9, 2, …} Why or why not?

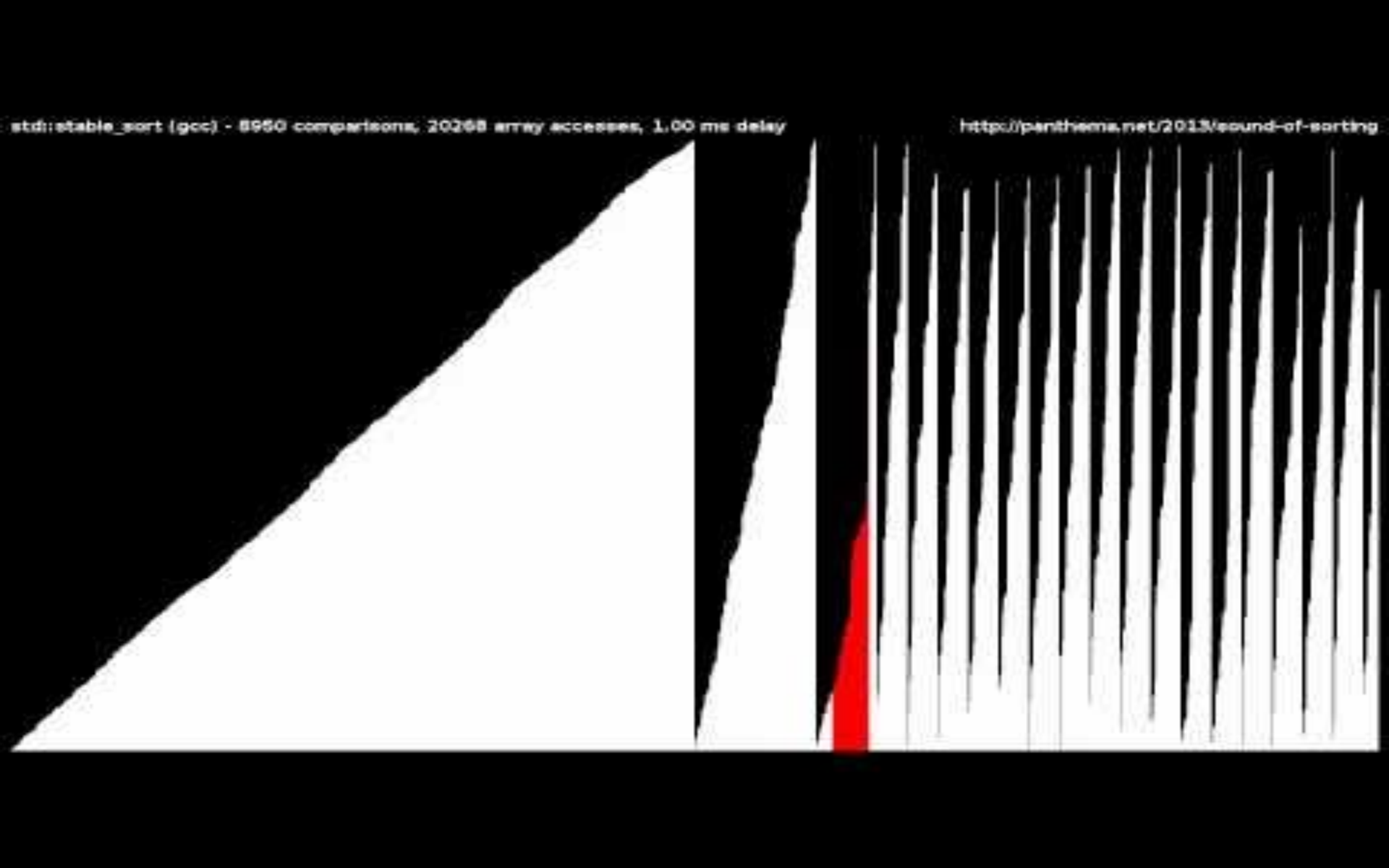
How can I make it "more sorted"?

Proposal - Bubble Sort:

Repeat for *n* times (where n is the total # of elements):

Iterate through all elements, and at the i-th location:

If element i and element (i+1) were out of order: swap them


std::stable_sort (gcc) - 8950 comparisons, 20268 array accesses, 1.00 ms delay
http://panthema.net/2013/sound-of-sorting

What's Next ...

"The Language of Mathematics": *Counting* and *measurement* are two different human-cognitive conceptions, upon the formulation of which numbers arose (Devlin, 2012, 4.2). "While logic provides a useful model of mathematical reasoning … in adopting this broader, societal perspective, I believe this book still help negotiate the transition from high school (to university mathematics) and also anyone develop better reasoning skills." (Preface)

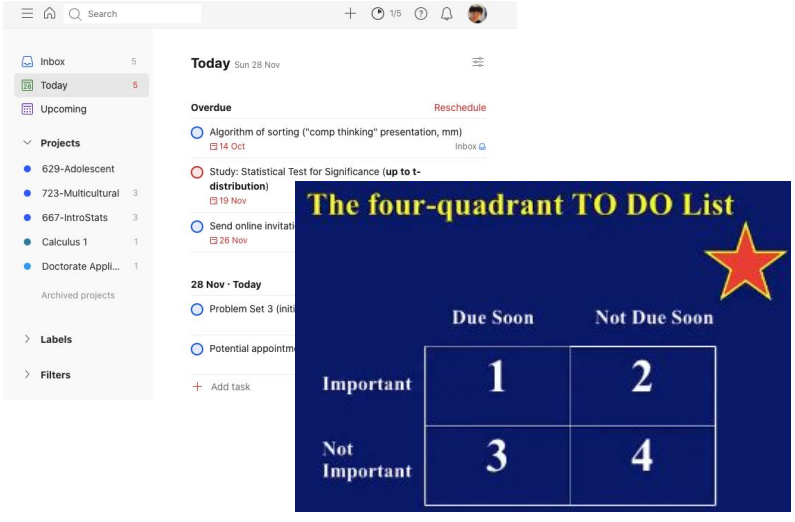
Asymptotic Notations

Models of Computation

Algorithm Design "Philosophical" Paradigms

- Incremental Construction,
- Iterative Refinement,
- Divide and Conquer, and more..

“Time Management”



(Carnegie Mellon, 2008)

- Proposed by American educator Stephen Covey, “The Eisenhower Matrix”
- We can always change the plan, but only if and once you *have* one!
- Try to be on-time; learn to say “No”
- “*Procrastination is the thief of time.*”

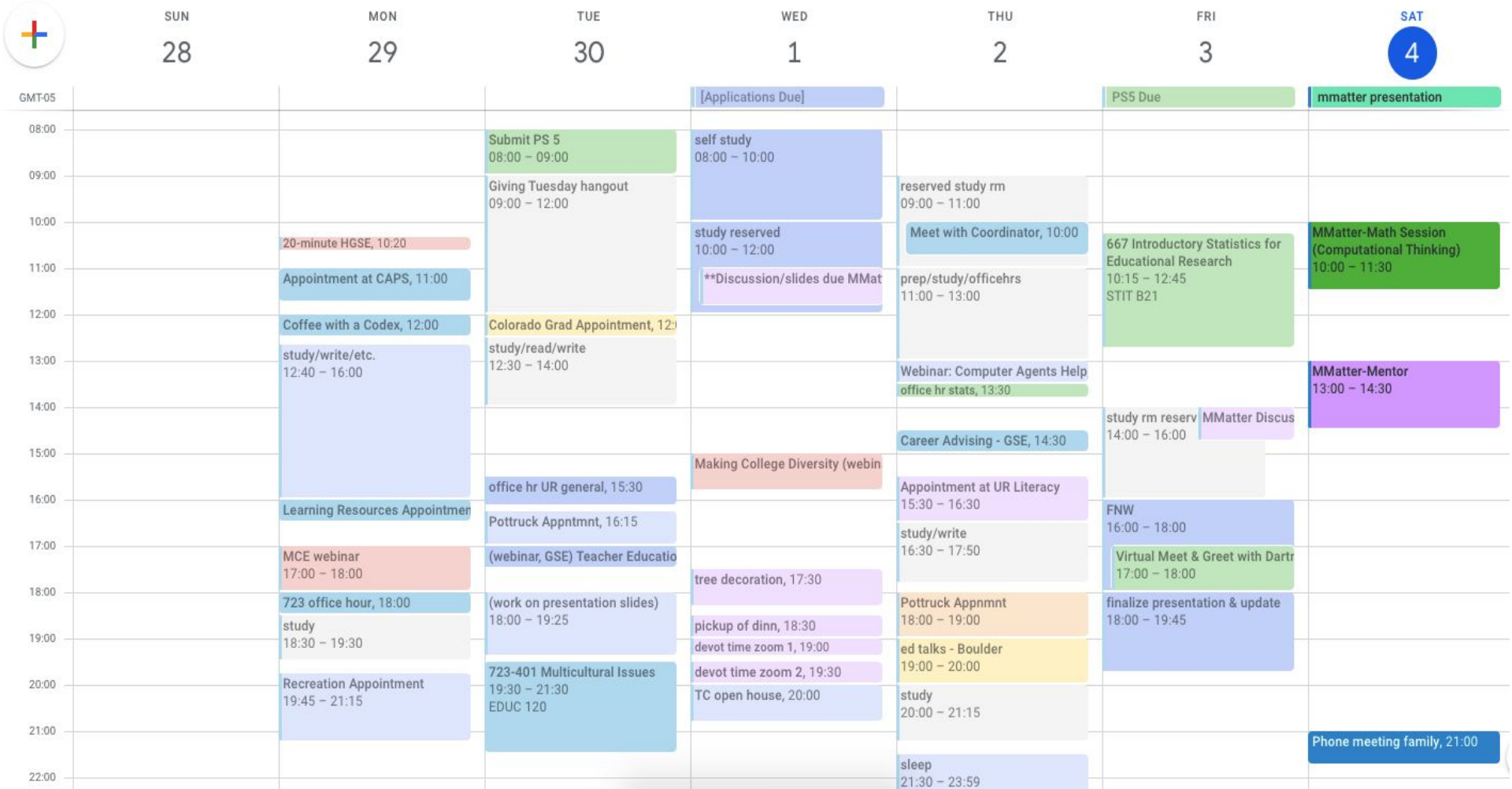
  Edward Young - Night Thoughts, 1742

Important and Urgent	Important but NOT Urgent
☐ Email for questions in course X	☐ Problem Set 3 (initial attempt)
☐ Synthesis paper abstract (Wed.)	☐ Send a few internship applications
☐	☐ Tutoring session preparation
☐	☐

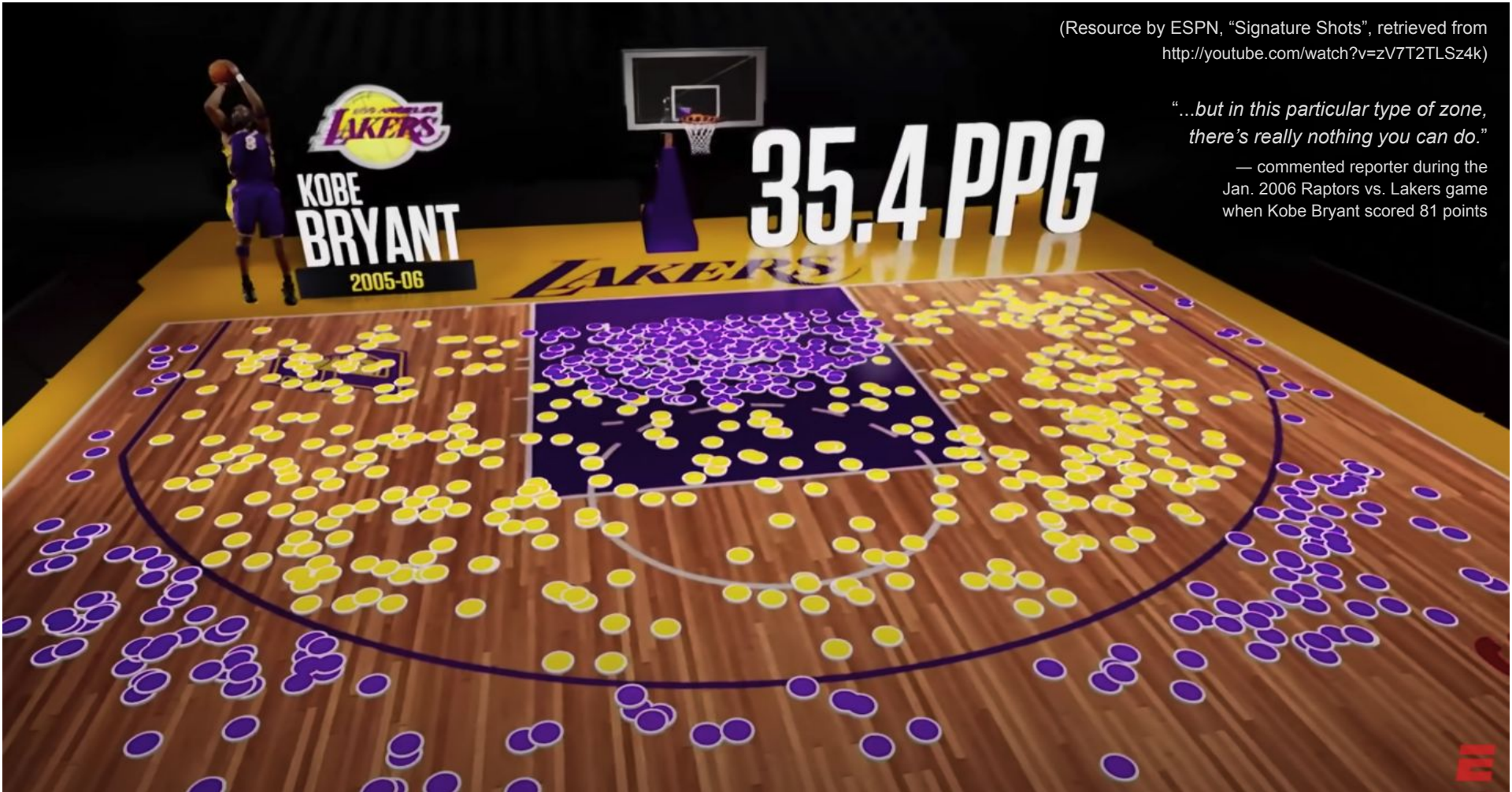
NOT Important but Urgent	NOT Important and NOT Urgent
☐ Validate carrel and rm reservations	☐ Grocery shopping for ingredients
☐ “Video of ideas” self-study assign.	☐ Read about other grad programs
☐ Make additional gym appointments	☐ Vacuum bedspace
☐ Follow up with literacy instructor	☐ Consider visiting arts library
☐ Send email to volunteer organization	☐ Practice plan for reading strategies
☐ Resume cal one course online	☐ Wrap up intro stats lecture notes
☐ Submit synthesis paper abstract	☐ Return a few library books
☐ ...	☐ Find used bookshelf for apartment
☐	☐ Travel plans for winter break
☐	☐ ...
☐	☐
☐	☐
☐	☐

(Academic Success & Disability Services, University of Redlands, 2015. Retrieved from http://www.redlands.edu/globalassets/depts/student-affairs/academic-success/skills-worksheets/4-quadrant_to-do_list_updated_2015.pdf)

“Time Management” (cont.)



Revisited: Basketball Jumpshots Vis. Design



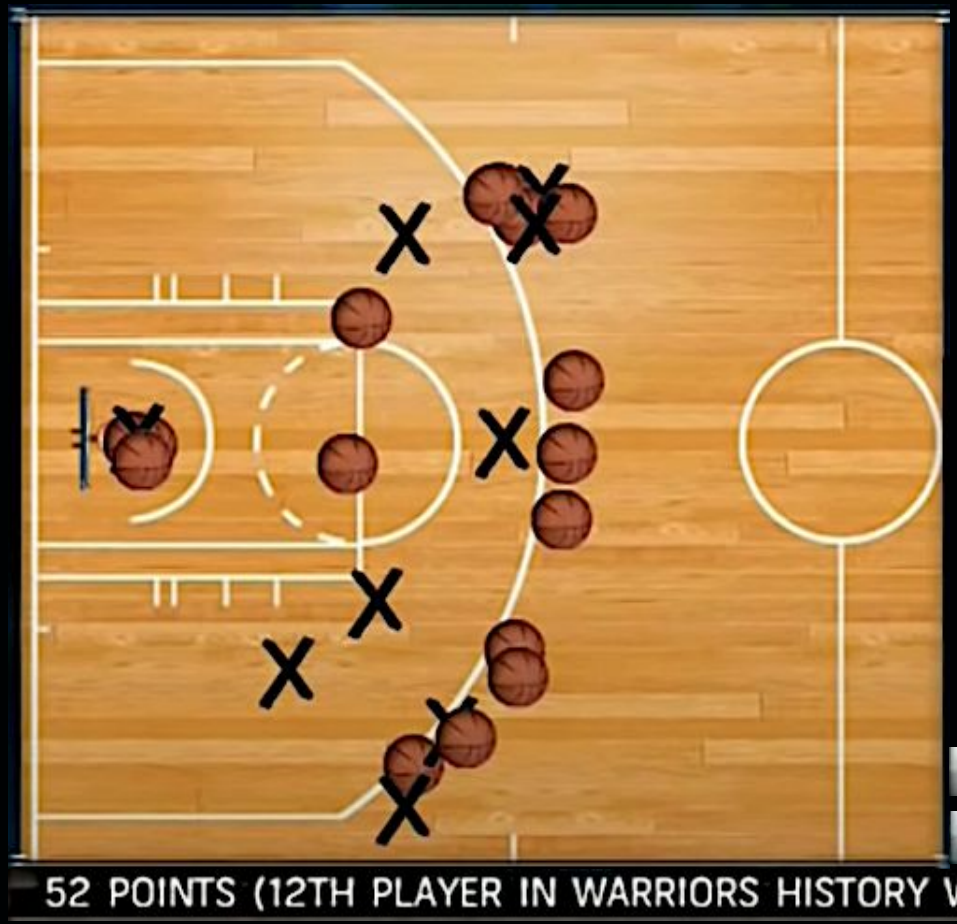
(Resource by ESPN, “Signature Shots”, retrieved from http://youtube.com/watch?v=zV7T2TLSz4k)

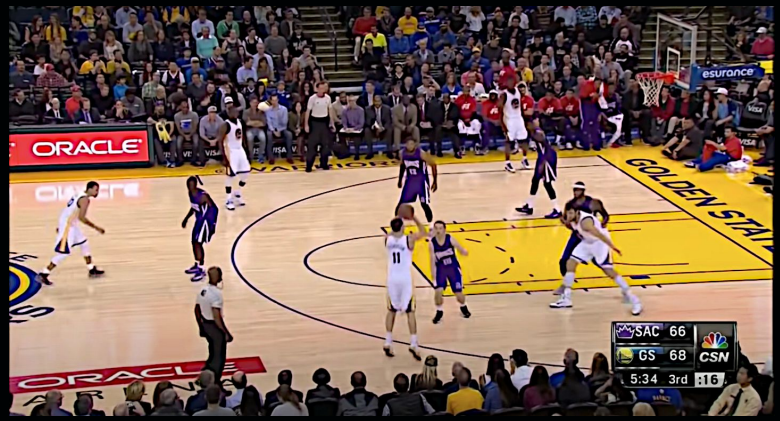
“...but in this particular type of zone, there’s really nothing you can do.”

— commented reporter during the Jan. 2006 Raptors vs. Lakers game when Kobe Bryant scored 81 points

Revisited: Basketball Jumpshots Vis. Design (cont.)

(Resource by Comcast SportsNet/CSN, retrieved from http://www.youtube.com/watch?v=5nyBpt9tRsg)





(Youtube, Klay Thompson 37pt 3rd Quarter CSN Bay Area feed 1-23-15)

References and Suggested Reading

Devlin, K. (2012). Introduction to Mathematical Thinking. *Keith Devlin*, Stanford University (http://profkeithdevlin.com).

Devlin, K. and Stanford University (on-going). Introduction to Mathematical Thinking. Coursera. (URL).

Borowski, D. (2021). The 10 Most Popular Coding Challenge Websites. *freeCodeCamp* (URL).

Project Euler (Tue, 30 Nov 2021). About Project Euler. http://projecteuler.net

Cormen et al. (2009). Introduction to Algorithms, 3rd Edition. *The MIT Press*.

Stroustrup and AT&T Bell Laboratories (1994). The Design of C++. *Computer History Museum (Youtube)*. (URL).

Carnegie Mellon (2008). Randy Pausch Lecture: Time Management. *(Youtube)* https://www.youtube.com/watch?v=oTugjssqOT0

Pausch and Zaslow (2008). “Managing Your Time: Words of Advice” from The Last Lecture. *Fast Facts*. Used by permission of Hyperion. All rights reserved (URL).

Thank You!

dsun2019@hotmail.com