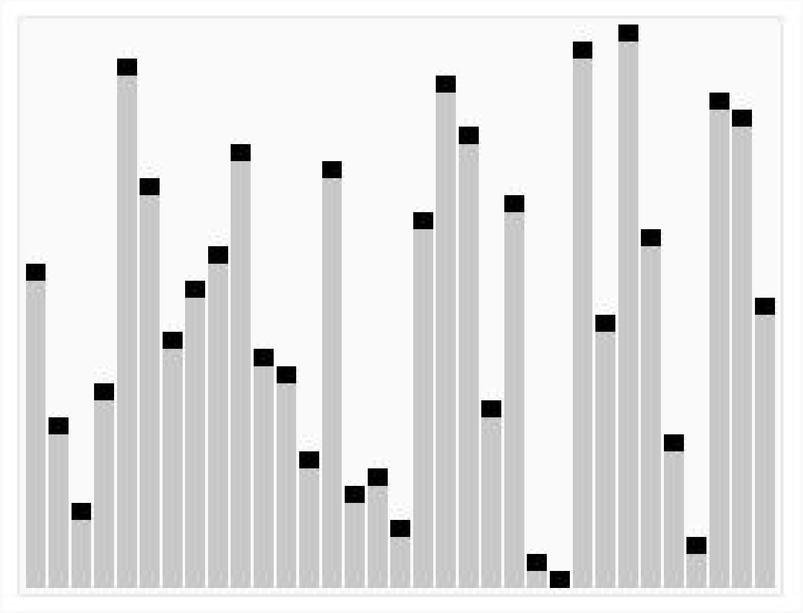
Sorting #4 -
One Sort to Rule Them All
One Sort to Find Them

CptS 223 - Fall 2017 - Aaron Crandall

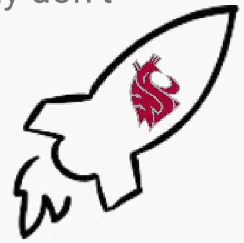
Today’s Agenda

- Announcements
- Thing of the day
- Time for some quicksort



Announcements

- MA 5 - Instrumented sorting is due to go out
  - Total code is… not incredibly long, but the instrumentation could take a bit
- HW3 is out
  - If you do it on paper, please BOTH scan & commit it to the HW3 branch of your repository *AND* bring the paper copy into class. The TA's are asking for anything where you've written by hand because it's tough to grade from some of the scans
  - In fact, if you have HW1 or HW2 on paper, please bring those in too if you already don't have a grade for them.
- Today & Tomorrow - Cougs in Space days! Very cool stuff
- I might need to duck out of office hours early, not sure yet



PROGRAM AGENDA

WEDNESDAY, 10/25

Information Sessions in ETRL 101	
4:00-4:45	Planetary Resources
4:45-5:30	Blue Origin
5:30-6:15	Systima Technologies
6:15-7:00	Raytheon

THURSDAY, 10/26

9:00-11:30 Lighty 160	Student Interviews or Drop-in Office Hours
12:00-1:00 CUB 204	Student Lunch with Industry (No-Host)
4:00-6:00 CUE 203/ CUE Atrium	Panel Discussion and Networking Reception

PARTICIPANTS

RYAN CROMPTON Outreach Coordinator, Blue Origin

In fall 2017, Ryan Crompton joined Blue Origin as an Outreach Coordinator focusing on University and Technical College outreach. Before joining Blue Origin, Crompton worked at Boeing for 5 years in the Autoflight and Avionics flight test and certification group.

RONALD BLIESNER Fluid Systems Design Engineer, Blue Origin

Proud WSU Alumni, Ron Bliesner has worked at Blue Origin for the past four years, where he has done everything from initial system conceptual design to final integration. He has designed, built, and tested cryogenic systems and components for both the New Shepard and New Glenn vehicles.

JUSTIN BAHRAMI Engineer, Blue Origin

Justin Bahrami earned his B.S. degree in Mechanical Engineering from Washington State University in December 2012. Bahrami worked on cryogenic systems while at WSU, including the Genii UAS, an aircraft that is intended to be powered by liquid hydrogen and a PEM fuel cell.

ERIC FRAISER Talent Miner (HR), Planetary Resources

At Planetary Resources, Eric Fraiser leads Recruiting and has additional duties in overall staff management. He has recruited for technology research and product development teams at Apple, Google, HTC, Oracle, and Microsoft as well as a long list of technology start-ups and industry casualties.

PETER ILLSLEY Director of Mech. and Thermal Eng, Planetary Resources

Peter Illsley joined Planetary Resources from the NASA Jet Propulsion Lab where he was the lead integration engineer for the Curiosity Mars Rover and the last human to touch Curiosity. He was in Mission Control during the "6 minutes of terror" landing sequence, waiting for the rover to activate.

HUNTER GOLDEN Chief Engineer, Systima Technologies

Golden has 23 years of experience in research, development, and test of aerospace and defense systems. As the chief engineer, Golden is responsible to ensure the complete and consistent execution of engineering design, development and testing the quality of pyrotechnic system design.

JOHN SCHLAERTH Sr. Principal System Eng./Manager III, Raytheon Company

John Schlaerth is an electo-optical space hardware design manager and the cryogenics product line leader. He is active in business pursuits in designing and manufacturing of next gen. earth observation payloads, and overseeing research and development of low exported torque pulse tube Cryocoolers.

Top Hacker (what does that even mean?)

- TEDxMidwest talk on various hacking things
- Includes lasers killing mosquitoes!

https://www.youtube.com/watch?v=hqKafI7Amd8

- Shows how secure new credit cards are (lolz)
- Amazingly secure locks (not!)
- Wifi networks are amazing (and privacy wrecking)
- Oh, and he is treating mosquitos as they need to be treated… with lasers!

Quicksort - Named (mostly) appropriately

- Runs in O(N log N) time
- Uses a divide and conquer strategy (like merge sort)
- Very sensitive to implementation
    - Pick a good pivot or you’re doomed
- Recursive algorithm
    - Can work in the single array, so no doubling of memory space
- Invented by Sir Tony Hoare in 1960 (Aged 26)
    - Speaking at a conference in 2009, he apologised for inventing the null reference

Performance of Quicksort

- Average running time: O(N log N)
- Worst case: O(N^2)
  - Can be made exponentially unlikely with a small tweak
  - All of the performance issues are centered around picking the pivot value
- Often combined with other sorts to improve efficiency:
  - When N (of any sublist) is 5 <= N <= 20 -> change to Insertion sort

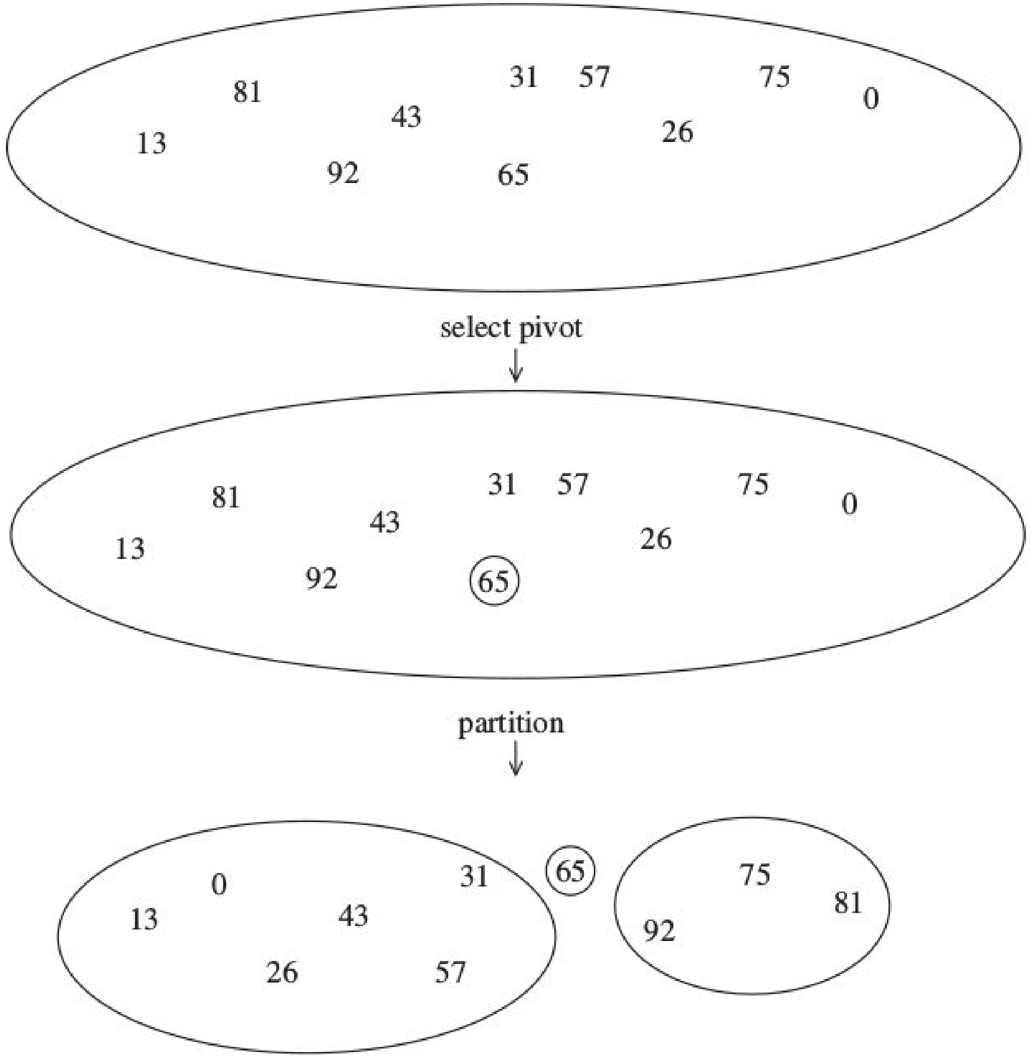
Basic Idea: Divide over pivot, repeat

- For list S:
  - If S.size() is 0 or 1, return
  - Pick element v in S, name it pivot
  - Partition S - {v} (take out the pivot) into two groups:
    - Those smaller than v -> S_1
    - Those bigger than v -> S_2
  - Return { quicksort(S_1), v, quicksort(S_2) }

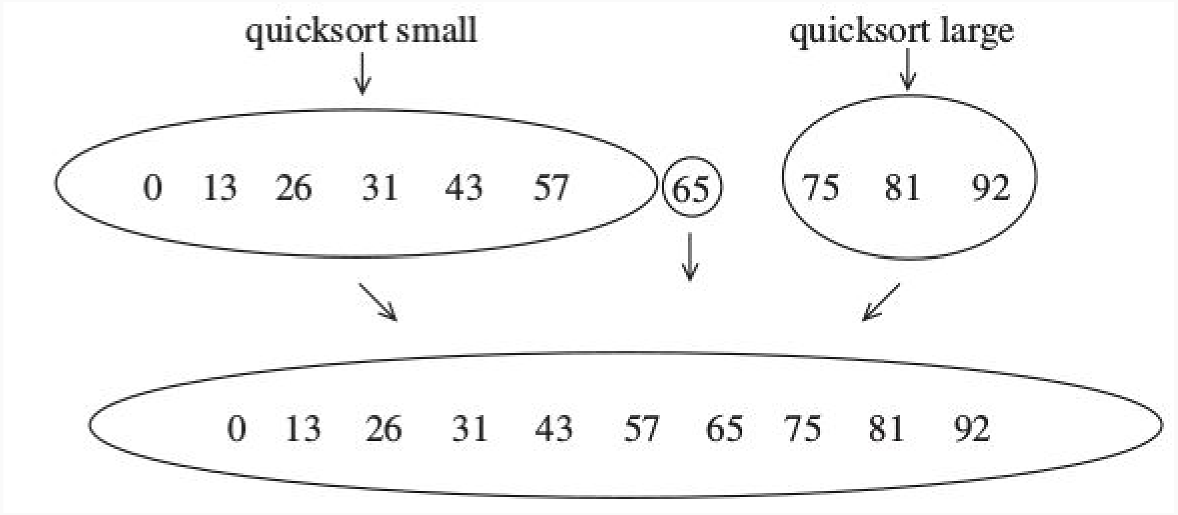
WIN

Divide values above and below pivot

- Pick pivot
- Divide into two lists
  - S_1 -> Those values < pivot
  - S_2 -> Those values > pivot



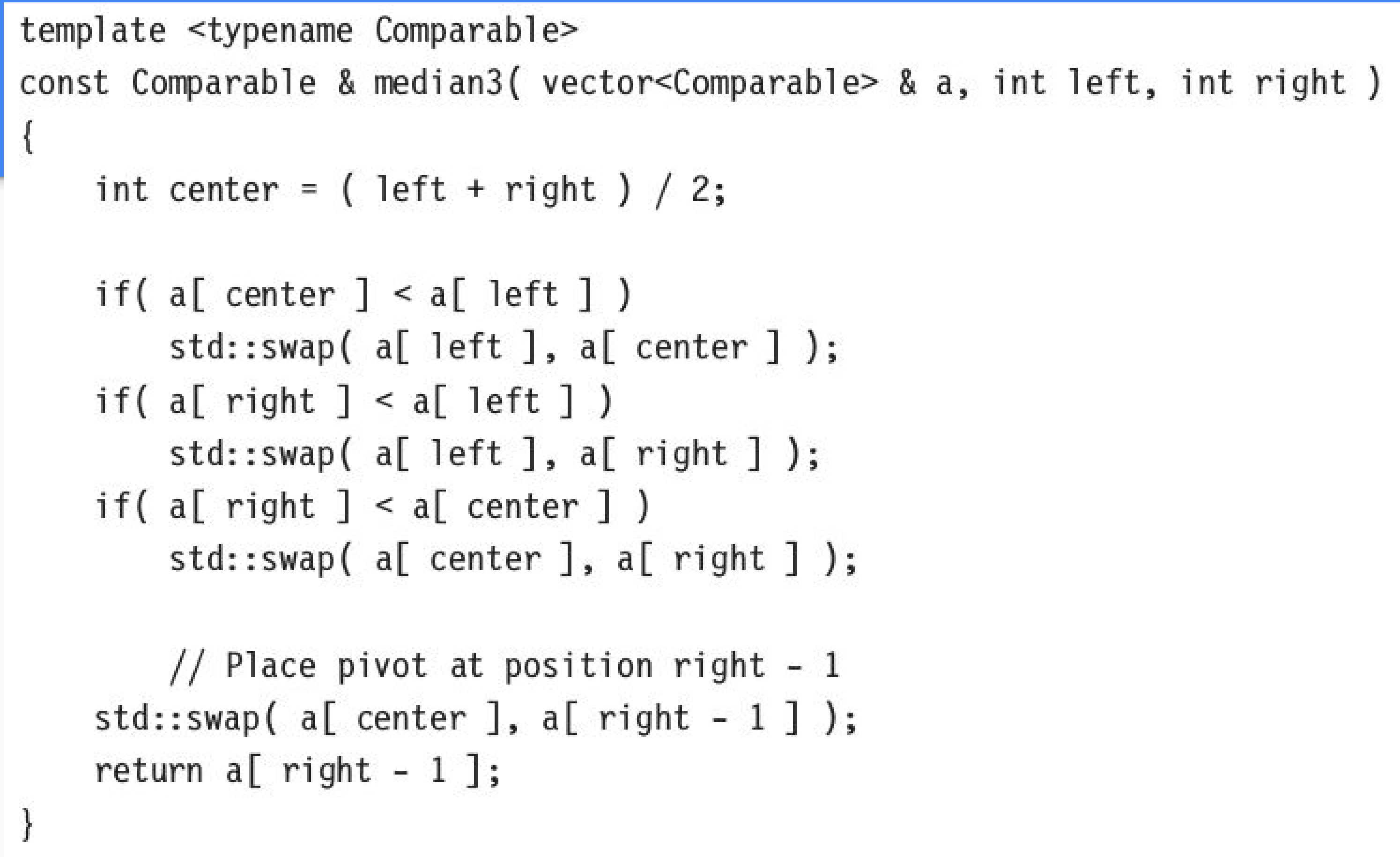
Then combine back together



Picking a pivot - What truly makes quicksort fast or slow

- Can go *horribly* wrong
- *Should* be median (physically middle value in list) of the list, right?
  - Finding median is very slow (O(N))
- Pick a random index? -> no real guarantee of any good value
- Choose first element? -> What if it’s sorted? (you get N^2 time)
  - BONUS: You also get stack depth == N for bonus memory consumption
- Choose last element? Ditto
- So… median of 3 seems to work in most cases - basically a sampling way
  - medianOf{First, middle (S.size() / 2), Last}

Median Finder



```
template <typename Comparable>
const Comparable & median3( vector<Comparable> & a, int left, int right )
{
    int center = ( left + right ) / 2;

    if( a[ center ] < a[ left ] )
        std::swap( a[ left ], a[ center ] );
    if( a[ right ] < a[ left ] )
        std::swap( a[ left ], a[ right ] );
    if( a[ right ] < a[ center ] )
        std::swap( a[ center ], a[ right ] );

        // Place pivot at position right - 1
    std::swap( a[ center ], a[ right - 1 ] );
    return a[ right - 1 ];
}
```

Visually

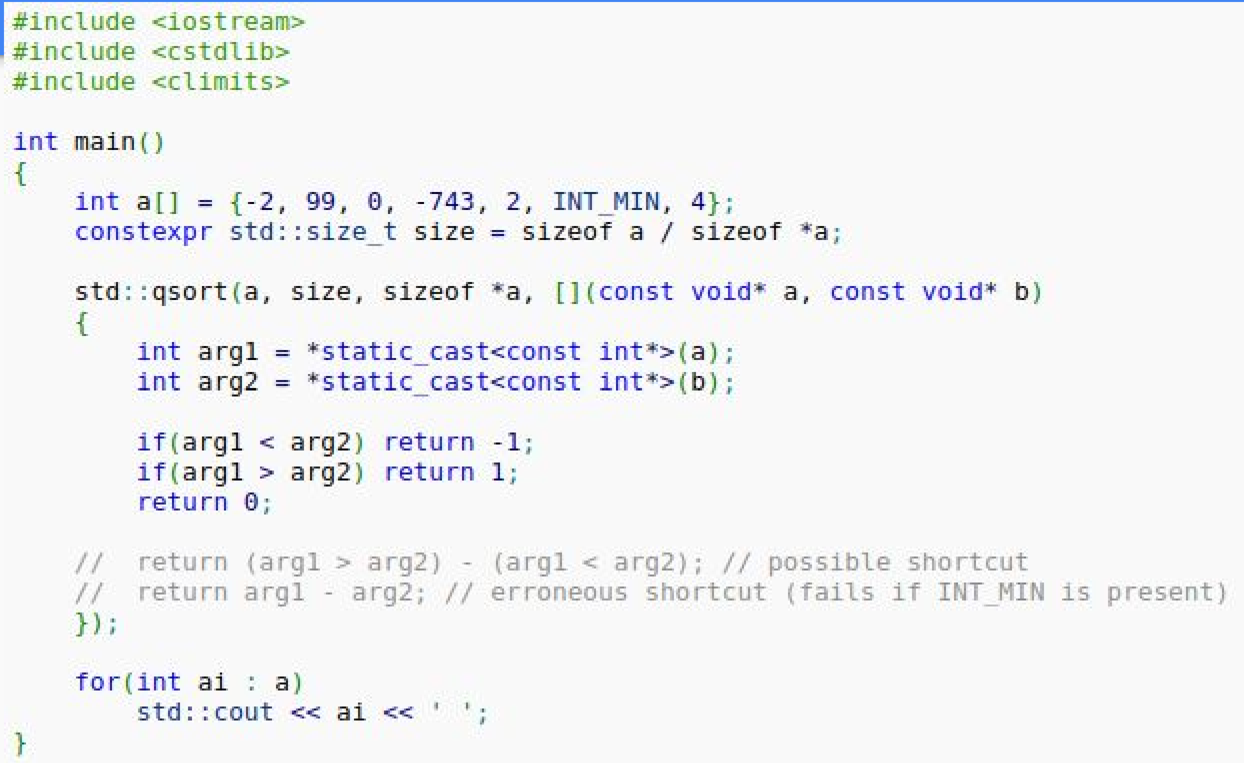
- https://visualgo.net/sorting

- A few key ones: https://www.youtube.com/watch?v=ZZuD6iUe3Pc
- High speed! https://www.youtube.com/watch?v=BeoCbJPuvSE

C++11 Quicksort: qsort

http://en.cppreference.com/w/cpp/algorithm/qsort

- C++11 STL has a Quicksort function via cstdlib



```
#include <iostream>
#include <cstdlib>
#include <climits>

int main()
{
    int a[] = {-2, 99, 0, -743, 2, INT_MIN, 4};
    constexpr std::size_t size = sizeof a / sizeof *a;

    std::qsort(a, size, sizeof *a, [](const void* a, const void* b)
    {
        int arg1 = *static_cast<const int*>(a);
        int arg2 = *static_cast<const int*>(b);

        if(arg1 < arg2) return -1;
        if(arg1 > arg2) return 1;
        return 0;

    //  return (arg1 > arg2) - (arg1 < arg2); // possible shortcut
    //  return arg1 - arg2; // erroneous shortcut (fails if INT_MIN is present)
    });

    for(int ai : a)
        std::cout << ai << ' ';
}
```

Easier to use the std::sort interface

- std::sort(s.begin(), s.end());

Dancing!

Insertion Sort: https://www.youtube.com/watch?v=ROalU379l3U

Quick Sort: https://www.youtube.com/watch?v=ywWBy6J5gz8

Merge Sort: https://www.youtube.com/watch?v=XaqR3G_NVoo

Bubble Sort: https://www.youtube.com/watch?v=lyZQPjUT5B4

For Friday: Radix / Counting sorts! These are non-comparison sorts

- We can get O(N) sorting time, given the right kind of conditions…
- Next week:
  - Monday: Probably Valgrind.
  - Wednesday: Midterm review
  - Friday: Midterm #2 - Heaps, Hashing, Sorting