Lab 4:

You may work in a group of up to 3 people if you want. If you choose to do this, ensure that you put the name of all team members into your lab6.md file. Working together can help you understand the content of the videos. You are all expected to watch the video and be part of the discussion. **Each member must submit their own copy of the work but the answers for part A can be the same. Reflection needs to be independently done**

1. Watch the following video: Sorting Algorithms: Speed Is Found In The Minds of People - Andrei Alexandrescu - CppCon 2019

2. Answer the questions in the next section, You can use the internet to help you learn about ideas that you are unfamiliar with, but you will need to link and cite your sources of information. A link is provided that may be useful for you at bottom of this lab.

3. Write a reflection (this part is individual)

Part A: Questions about video

Do not forget to add names (and email addresses) of all group members to top of file if you are working in a group.

1. What sorting algorithm was the speaker trying to improve?

2. At what partition size does VS perform a simpler sort algorithm instead of continuing to partition?

3. At what partition size does GNU perform a simpler sort algorithm instead of continuing to partition?

4. Regular insertion sort does a linear search backwards from end of array for correct spot to insert. According to the speaker, why does switching to a binary search not improve performance?

5. Describe what is meant by branch prediction? (this may require further research)

6. What is meant by the term **informational entropy**? (this may require further research)

7. If size == 15, what is size & 1? if size == 16, what is size & 1? Explain how right = first + 1 + (size & 1) avoids a conditional check. Hint:

  - The & is the bitwise AND operator in C/C++. It takes the bit representation of the two operands and perform an AND operation on each of the corresponding bits to form a result
  - To do this question first convert 15, 16 and 1 to base 2 (use 5 digit representation for all of them). Then perform an AND operation of the correseponding bits of the operands... this will get you a 5 digit binary value. Convert the value back to base 10 .

8. Speaker suggests the following algorithm:

   - make_heap()
   - unguarded_insertion_sort()

   He suggests that by doing make_heap() first then you can do something called unguarded_insertion_sort(). Explain what is unguarded_insertion_sort() and why it is faster than regular insertion sort. How does performing make_heap() allow you to do this?

9. The speaker talks about incorporate your conditionals into your arithmetic. What does this mean? Provide an example of this from the video and explain how the conditional is avoided.

10. The speaker talks about a bug in gnu's implementation. Describe the circumstances of this bug.

11. The speaker shows several graphs about what happens as threshold increases using his new algorithm. The metric of comparison is increased, the metric of moves are increased but time drops... What metric does the author think is missing? Describe the missing metric he speaks about in the video. What is the metric measuring?

12. What does the speaker mean by fast code is left leaning?

13. What does the speaker mean by not mixing hot and cold code?

Part B: Reflection

This part must be individually done.

1. What did you/your team find most difficult to understand in the video?
2. What is the most surprising thing you learned that you did not know before?

3. Has the video given you ideas on how you can write better/faster code? If yes, explain what you plan to change when writing code in the future. If no, explain why not.

References:

You may find these articles on branching and cache useful:

https://en.algorithmica.org/hpc/

Submitting your lab

Please provide all your answers in a single file emailed with subject: dsa-456-lab5