CSE 504: Assignment 2
Memory Access Frequency Analysis
Due date: Friday, November 3

In this assignment, you will study the analysis and optimization phase of the LLVM/Clang compiler and implement a pass to analyze the memory access frequency of data pointers. This assignment is intended to give you some insight into real-world compiler analysis and optimization and help you better understand the LLVM IR.

To complete this assignment, you will perform the work described below. The report and the files are to be made available to the TA for evaluation by midnight on the due date. Do **not** delete interim versions of your files or anything else that documents how you have performed your assignment tasks.

Here is what you should do:

For this work you will use your account on Seawulf and the version of LLVM/Clang that you have already installed. This assignment does not depend on the first assignment, but you will need all your previous code in the final assignment, so do not delete your files.

The things you need to accomplish in this effort are listed as follows.

1) Follow his link https://www.cs.cornell.edu/~asampson/blog/llvm.html to learn the basics about LLVM passes and follow the instructions give to prepare a skeleton pass. You should already have some basic knowledge about LLVM IR from Assignment 1, e.g., the format of load/store instructions. A more complete LLVM IR description can be found here: https://llvm.org/docs/LangRef.html. You can use it as a reference manual.

2) Identify load/store instructions in the program, and get execution frequency information from BlockFrequencyInfoWrapperPass for each load and store.

BlockFrequencyInfoWrapperPass provides information on the execution frequency of basic blocks. You should read its source code to get a basic understanding of how it works and implement code to leverage this information for memory access frequency analysis. A description of BlockFrequencyInfoWrapperPass can be found here: https://llvm.org/docs/BlockFrequencyTerminology.html.

3) Compute the read and write access frequency for pointer type function arguments. Identify addresses related to this type of arguments, and combine all their load/store access frequency.

4) Write your own test program and use the implemented pass to print the access frequency information you calculated.

As always, do **not wait** until the due date to do your work: otherwise the machine may become very full around the due date.

You will need to submit your files. Please also leave suitably named files in an appropriately named directory in your course account in case we need to inspect them. Your documentation should provide any explanations needed. Please only submit the pass file that you wrote. You should not modify any existing LLVM/Clang file in this assignment. Please send the file to the TA by midnight on the due date.

Please also submit a report for your work. Your report should describe the implementation of your memory analysis pass and the test results. It should be no more than 4 letter-size pages in length. Email your report in pdf or word format to the TA by midnight on the due date. You are required to submit a **hard copy** of the report on the Monday after the due date.

Grading:

Grades from A through F will be assigned. Grades will be based upon the correctness of your work, completeness of your responses, and the overall quality of your work. Please do your own work: do NOT copy code or text from other class participants or from any other source.