CS 341 – Assignment #4
Due 15-June-2011

1 Problem 1

Due to branch mispredictions, for small values of n a $O(n)$ linear search may be faster than an $O(log_2 n)$ binary search. Using the supplied binary search and linear search code, experimentally determine the "break even point." This is the value of n for each the two search algorithms have approximately the same performance. The time used by the routines can be measured using the `__rdtsc`[1] function.

For your results to be meaningful you must perform multiple tests for each value of n. At the very least, be sure to hit the best case, the worst case, and several "average cases" for each search algorithm.

In addition, perform separate measurements for the hot and cold cache cases. The arrays in question are small enough to fit completely in the CPU L1 cache. The hot cache case can be tested by using the linear search for the last element in the array (this will access every element in the array thereby fetching it into the cache).

Be sure to compile your code in "release" mode. Compiling in "debug" mode, which is the default, disables almost all optimization done by the compiler.

You will turn in:

- The single `.c` or `.cpp` file containing your measurement code.
- A graph of the "cache hot" times for the binary search and the linear search at various values of n. The x-axis should be n, and the y-axis should be average number of CPU cycles.
- A graph of the "cache cold" times for the binary search and the linear search at various values of n. The x-axis should be n, and the y-axis should be average number of CPU cycles.

For the graphs, I would prefer a `.PDF` of just the graph. However, I will also accept a `.ODS`, `.XLS`, or `.XLSX` of the graph with the raw data. If the graphs are submitted as spreadsheets, the two graphs should be separate sheets in the same file. If the graphs are submitted as `.PDF`s, the may either be separate files or separate pages in one file. Submit the three (or two) files specified in a single `.ZIP` file named `lastnameFirstname-assignment4.1.zip`.

Extra credit: In the binary search, the difference between `hi` and `lo` will eventually fall below the experimentally determined threshold where the linear search is faster. Modify the binary search routine to detect this case and call the linear search routine instead. Call this new routine `binary_and_linear_search`. Perform the same measurements on this code and include the graphs.

2 Problem 2

Consider the following C implementation of matrix multiplication.

```
void matrix_multiply(const float *a, const float *b, float *c, unsigned dim)
{
        unsigned i, j, k;

        for (i = 0; i < dim; i++) {
                for (j = 0; j < dim; j++) {
                        double accum = 0.0;
                        for (k = 0; k < dim; k++) {
                                accum += a[(i * dim) + k] * b[j + (k * dim)];
                        }
                        c[(i * dim) + j] = accum;
                }
        }
}
```

[1] `http://msdn.microsoft.com/en-us/library/twchhe95.aspx`

This code correctly implements the matrix multiplication, but for large values of `dim`, the performance is very bad. Modify this routine to perform better for large arrays. The optimized version must work for all values of `dim`, and it must produce results within ϵ of the original routine.

You will turn in:

- The single `.c` or `.cpp` file containing your optimized matrix multiplication routine.
- A graph of the execution times for the original function and the optimized function for various matrix sizes from 64×64 to 5120×5120 (yes, that is 300MB of matrices).
- A short paragraph describing how you would further optimize this routine to handle matrices as large as 32768×32786. That is 12GiB of matrices, and is too large to fit in memory. At this size, TLB misses and paging begin to become a serious issue. How would these be addressed? This paragraph should be in a plain text (`.TXT`) file. I do *not* want a `.DOC` or `.DOCX` file.

For the graph, I would prefer a `.PDF` of just the graph. However, I will also accept a `.ODS`, `.XLS`, or `.XLSX` of the graph with the raw data. If the graphs are submitted as spreadsheets, the two graphs should be separate sheets in the same file. If the graphs are submitted as `.PDF`s, the may either be separate files or separate pages in one file. Submit the three files specified in a single `.ZIP` file named `lastnameFirstname-assignment4.2.zip`.