CS121 Problem Set 2

Due date: Before 7pm on April 4, 2019. Email your solutions to the TA 官加文 (guanjw@shanghaitech.edu.cn) or submit a paper copy to him before the due date.

1. Consider the following sequential *rank sort* algorithm for n values assuming no duplicate values:

```
for (i = 0; i < n; i++) {
    x = 0;
    for (j = 0; j < n; j++)
        if (a[i] > a[j]) x++;
    b[x] = a[i];
}
```

(a) Rewrite this as a parallel algorithm using OpenMP assuming that $p < n$ threads are used. Indicate clearly the use of private and shared variables and the schedule type.

(b) Modify the OpenMP code in part (a) to handle duplicates in the list of values, i.e. to sort into non-decreasing order. For example, the list of values [3, 5, 7, 5, 7, 9, 2, 3, 6, 7, 8, 1] should give the sorted list [1, 2, 3, 3, 5, 5, 6, 7, 7, 7, 8, 9].

2. The *back substitution* algorithm solves a set of linear equations in upper (or lower) triangular form, as shown below. Design a parallel algorithm for back substitution for a shared memory architecture and express it using OpenMP, assuming that $p < n$ threads are used.

$$
\begin{aligned}
a_{n-1,0}x_0 + a_{n-1,1}x_1 + a_{n-1,2}x_2 \quad \cdots \quad + a_{n-1,n-1}x_{n-1} &= b_{n-1} \\
&\vdots \\
a_{2,0}x_0 + a_{2,1}x_1 + a_{2,2}x_2 &= b_2 \\
a_{1,0}x_0 + a_{1,1}x_1 &= b_1 \\
a_{0,0}x_0 &= b_0
\end{aligned}
$$

3. GPU computing has been used to speed up many real-world applications. However, not all applications are suitable for GPU acceleration. Consider the following operations / applications and decide whether each is suitable for GPU acceleration or not. Briefly justify your answer. Assume all the input data are initially in the main memory.

   a. Matrix multiplications on two matrices *A* and *B*, each of size 32000 by 32000.
   b. Matrix multiplications on two matrices *A* and *B*, each of size 32 by 32.
   c. A single binary search on a sorted array with one billion elements.
   d. Many binary searches on a sorted array with one thousand elements.

4. Compared to CPU threads, GPU threads are designed to be "light-weight". Describe some potential disadvantages associated with increasing the amount of work done in each CUDA thread (e.g. using loop unrolling) and thereby decreasing the total number of threads used.

5. You need to write a kernel that operates on an image of size 400 x 900 pixels. You would like to assign one thread to each pixel, and you want your thread blocks to be square with dimensions equal to a power of two. Suppose your device supports up to 1024 threads per block, up to 2048 threads per SM, and up to 16 thread blocks per SM.

   a. What size thread blocks should you use to maximize the number of threads on each SM?
   b. For the block size from part a, how many idle threads will you have?

6. Design a CUDA program to multiply an $n \times n$ matrix A by an $n \times 1$ vector B to produce another $n \times 1$ vector C, where $C[i] = \sum_{j=1}^{n} A[i][j] \cdot B[j]$. Use one thread to produce each value in C.