Math 590 Extra Credit Fall 2019

Due: Tuesday, November 5th

1. **Dwarves on a GPU**

   (a) **Barrel out of Bond**

   The Story:
   You live in a mystical world, where you are the proud possessor of magical ring of invisibility, and the companion of dwarfish royalty. You are traveling with the royal couple and their retinue, and together you are returning home.

   While passing through the binary forest your party is ambushed by hostile forest elves, and taken prisoner. All of the dwarves are captured, but you slip on your ring and avoid getting caught. The dwarves are locked up in a dungeon in the elves' forest stronghold. After several weeks of wandering around you have found where the elves keep the keys. Your problem now is how to get the dwarves out of there. A group of unarmed dwarves cannot simply walk out the front door and expect to get away.

   After a few more days you discover that under the dungeons the elves have a room full of empty barrels sitting next to a river. The barrels come in full of food through the front door. When they are emptied, the elves roll them into the river, and the barrels float in the swift current far away from the binary forest.

   This gives you a plan. You free the dwarves and get them to the barrel room. Then each dwarf gets into a barrel, with a final barrel for you, and you all will float away on the river. But the problem is this: to get a barrel into the river, one person (you or a dwarf) has to hold the barrel still in the water while another person gets in. You are more agile than a dwarf, so you can get in your barrel alone. But each of your dwarven companions will need assistance.

   The guards are on their way, and you need to escape as fast as you can. What is the fastest way for you and the dwarves to enter the barrels?

   The Specifics:

   - There are m dwarves and you, for a total of $n = m + 1$ companions. There are at least n barrels, so you are guaranteed that everyone has a barrel.
   - Each dwarf requires the assistance of one other person (you or another dwarf) to enter a barrel. This takes about 10 seconds.
   - You are able to enter your own barrel in about 10 seconds.

   Describe an algorithm for this barrel boarding process that minimizes the (asymptotic) time it takes to escape. Recall that $10f \in O(f)$.

ANSWER:

Pair up everyone and help half of the dwarves into their barrels. Now only $n/2$ people are not in barrels yet (you wait until the end). Repeat this process until only you are left, and finally get into your own barrel. There will be $O(\log n)$ iterations.

(b) **Summation on a GPU**

A GPU is a shared memory, parellel environment, where every one of the thousands of threads has access to the same, single version of the stored data. While running, all of the threads read and execute the same code. However, each thread has a unique identifier, and parallel behavior can be achieved by referring to these identifiers (i.e., something like `if(ID == x)` or `array[x+ID]`). This allows us to essentially give each thread its own, unique task to perform simultaneously.

(i) Assume we have an array of length n. How long would it take to add 1 to the value of every element, if we were to do this **in serial**?
ANSWER: This would require a linear scan of every element, $O(n)$.

(ii) Assume we have an array of length n. How long would it take to add 1 to the value of every element, if we were to do this **in parallel** on a GPU? You may assume that you have more than n threads available.
ANSWER: Since we have n threads, assign each thread to add 1 to a corresponding element. This can all be done simultaneously, so $O(1)$ runtime.

(iii) Assume we have an array of length n. How long would it take to compute the sum of every element in the array, if we were to do this **in serial**?
ANSWER: This would require a linear scan of every element, $O(n)$.

(iv) Assume we have an array of length n. Describe an algorithm to compute the sum of every element in the array **in parallel** on a GPU. You may assume that you have more than n threads available.
ANSWER: We take the same approach as the dwarves! In $O(1)$ time, add the second half of the array into the first half (i.e., add location $n/2$ into location 0, location $n/2+1$ into location 1, etc). Repeat this process on the remaining half of the array. Continue until the overall sum is contained in location 0.

(v) What is the runtime of your algorithm?
Each iteration, we half the number of elements to sum. So $O(\log n)$.