ECE590/MATH560 Homework 6 Extra Credit

Libo Zhang.

1. Dwarves on a GPU

(a) Barrel out of Bond [1pt].

Solution: We have m dwarves and myself, a total of $n = m+1$ companions. First, for all m dwarves, divide them into two groups (A and B), each group has $(\frac{m}{2})$ dwarves. Each dwarf in group A should help one dwarf in group B hold the barrel still in the water, and this group B dwarf woud get in the barrel and get away. In such way all $(\frac{m}{2})$ dwarves can get away in 10 seconds. Then, divide group A into two group again, with $(\frac{m}{2^2} = \frac{m}{4})$ dwarves in each group. Repeat the previously mentioned "one group help the other group hold barrel" method, then another $(\frac{m}{4})$ dwarves can get away in the second 10s.

Keep dividing the group and repeating the "help hold barrel" method, until we reach $(\frac{m}{2^1}, \frac{m}{2^2}, \frac{m}{2^3}, \dots, \frac{m}{2^k} = 1)$ situation, this means only 1 dwarf and myself are left, after $10 \cdot \log_2 m = \lceil 10 \cdot k \rceil$ seconds. Then, I spend 10s helping this dwarf get away, and finally I spend another 10s getting myself away.

Therefore, after $(\cancel{10\log_2 m}\ \lfloor 10 \cdot \log_2 m \rfloor + 10 + 10)$ seconds, all $n = m+1$ companions get away.

The asymptotic runtime of my algorithm is

$$10 \cdot \log_2 m + 10 + 10 = \left[10 \cdot \log_2 (n-1) + 20\right] \in O(\log_2 n).$$

(b) Summation on a GPU.

Solution (i). Assume one addition "+" operation takes constant [redacted] $\boxed{x}$ time, it would take $\cancel{\boxed{n \in O(n)}}$ $x \cdot n \in O(n)$ time, if we were to do this in serial.

(ii). If we were to do this in parallel, and we have more than n threads available, then it would take $x \in O(1)$ time.

(iii). In serial, we have $1 \underset{1}{+} 2 \underset{2}{+} 3 \underset{3}{+} 4 \underset{4}{+} 5 \underset{5}{+} \cdots + (n-2) \underset{n-2}{+} (n-1) \underset{n-1}{+} n$,

~~it would take~~ $\cancel{\boxed{(n-1)}}$ ~~$\in O(n)$ time.~~

this would take $x \cdot (n-1) \in O(n)$ time.

(iv). The algorithm is described in the following diagram.

① First addition

Use $(\frac{n}{2})$ threads for ①

$a_1 = 1+2$
$a_2 = 3+4$
$a_3 = 5+6$
$a_4 = 7+8$
$\vdots \quad \vdots$
$a_{\frac{n}{2}} = (n-1)+n$

② Second addition

[redacted] $a_1 + a_2 = b_1$
$b_2 = a_3 + a_4$
$b_3 = a_4 + a_5$
$\vdots \quad \vdots$
$b_{\frac{n}{4}} = a_{\frac{n}{2}-1} + a_{\frac{n}{2}}$

Use $(\frac{n}{4})$ threads for ②

③ Third Addition

$c_1 = b_1 + b_2$
$c_2 = b_3 + b_4$
$c_3 = b_5 + b_6$
$\vdots \quad \vdots$
$c_{\frac{n}{8}} = b_{\frac{n}{4}-1} + b_{\frac{n}{4}}$

Use $(\frac{n}{8})$ threads for ③

...... and move on

when we reach the k-th addition, ⓚ, where $\frac{n}{2^k} = 1$, $k = \log_2 n$, we finally have ~~our~~ finished computing the sum of every element, and get the final sum answer.

(v). According to (iv), the run time of my algorithm is

$$x \cdot k = x \cdot \log_2 n \in O(\log_2 n).$$