

1. Dwarves on a GPU

c.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, each group has $(\frac{m}{2})$ dwarves. Each dwarf in group A should help "A and B" one dwarf in group B hold the barrel still in the water, and this group B dwarf would 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 $(\frac{10 \cdot \log_2 m}{1} + 10 + 10)$ seconds, all $n = m+1$ companions get away.

$$\lceil 10 \cdot \log_2 m \rceil$$

The asymptotic runtime of my algorithm is

$$10 \cdot \log_2 m + 10 + 10 = \lceil 10 \cdot \log_2 (n-1) + 20 \rceil \in O(\log_2 n).$$

c) Summation on a GPU.

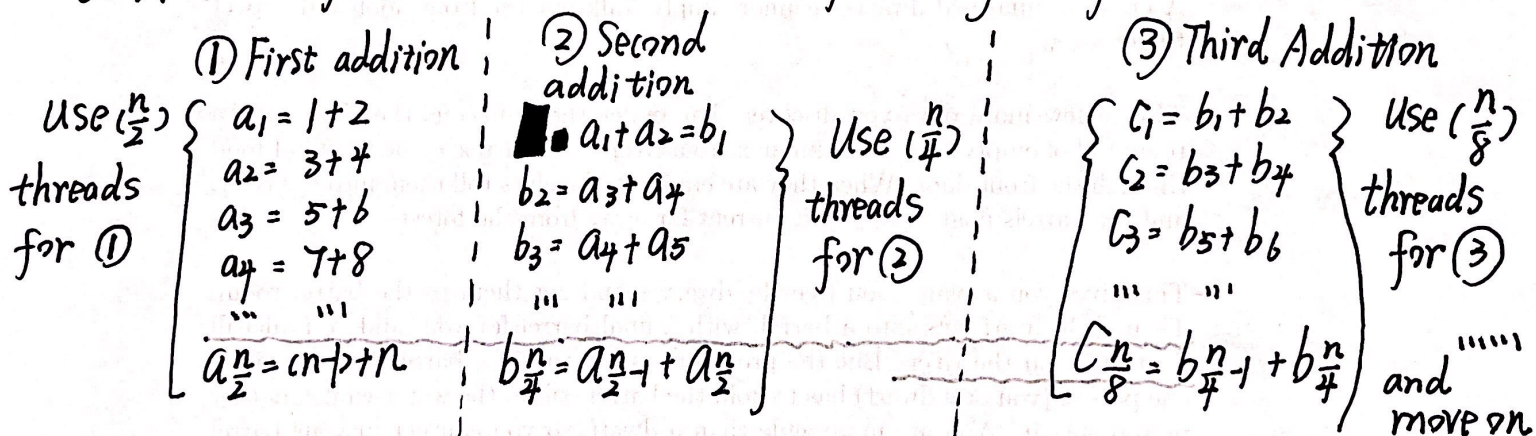
Solution (i). Assume one addition "+" operation takes constant λ time, it would take ~~$\lambda \cdot n \in O(n)$~~ $\lambda \cdot n \in O(n)$ time, if we were λ to do this in serial.

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

(iii). In serial, we have $1 + 2 + 3 + 4 + 5 + \dots + (n-2) + (n-1) + n$, it would take ~~$\lambda \cdot n \in O(n)$~~ $\lambda \cdot n \in O(n)$ time.

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

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



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

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

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