CX 4220/CSE 6220 High Performance Computing
Homework 1 Solutions

1. (a) Best case $= 1$; Worst case $= n$; Average case $= \frac{1}{n}\sum_{i=1}^{n} i = \frac{n+1}{2} \approx \frac{n}{2}$.

   (b) Best case $= 1$; Worst case $= \frac{n}{p}$; Average case $= \frac{1}{n/p}\sum_{i=1}^{\frac{n}{p}} i = \frac{\frac{n}{p}+1}{2} \approx \frac{n}{2p}$.

   (c) Best case speedup occurs when x is the first element in the last processor, and is $= n - \frac{n}{p} + 1 \approx n\left(1 - \frac{1}{p}\right) \to n$ for large p.
   Worst case speedup occurs when x is one of the first $\frac{n}{p}$ elements. Then, $T(n,1) = T(n,p)$ and speedup $= 1$.
   Average case speedup $\approx \frac{n/2}{n/2p} = p$.
   Observation: The speedup lemma stating $S(p) \leq p$ applies to worst-case complexity. For a specific instance which may turn out to be a bad case for the serial algorithm and a good case for the parallel algorithm, superlinear speedup is possible. In this problem, if the element to be searched is the first one on the last processor, the parallel algorithm will immediately find it while the serial algorithm has to walk through most of the array to get to the element.

2.
$$S(p) = \frac{n^2}{\frac{n^2}{p} + n}$$

   Scenario 1: When n is fixed and p is continually increased to its largest value n^2, the highest speedup obtainable is,

$$S(p = n^2) = \frac{n^2}{1+n} \to n = \sqrt{p} \qquad \{\text{for large } n\}$$

   Therefore, the speedup is bounded by $\sqrt{p}$.

   Scenario 2: When n is kept proportional to p, let $n = kp$ for some constant k,

$$S(p) = \frac{k^2p^2}{k^2p + kp} = \frac{p}{1 + \frac{1}{k}} = \Theta(p)$$

   This shows ideal speedup (and efficiency) can be obtained by increasing problem size commensurate with increase in number of processors. This corresponds to how parallel computers are used in practice (use larger systems to solve larger problems). This notion of speedup is called **Scaled speedup** (refer to *Reevaluating Amdahl's law* paper in Supplementary Reading).

3. Suppose run-time for the first is given by T_A and the second by T_B.

   Using Brent's lemma:

$$T_A(n,p) = \begin{cases} \frac{n^2\sqrt{n}}{p} & p \leq n^2 \\ \sqrt{n} & \text{otherwise} \end{cases}$$

$$T_B(n,p) = \begin{cases} \frac{n^2}{p} & p \leq n \\ n & \text{otherwise} \end{cases}$$

   Case 1: $p \leq n$
   Algorithm B is faster

   Case 2: $p \geq n^2$
   Algorithm A is faster

   Case 3: $n < p < n^2$

$$\begin{aligned} & T_A(n,p) \leq T_B(n,p) \\ \Rightarrow & \frac{n^2\sqrt{n}}{p} \leq n \\ \Rightarrow & p \geq n\sqrt{n} \end{aligned}$$

   Thus for $p < n\sqrt{n}$, algorithm B is faster and for $p > n\sqrt{n}$, algorithm A is faster.

4. (a) To be optimally efficient,

$$\begin{aligned} E(p) &= \Theta(1) \\ \frac{T(n,1)}{pT(n,p)} &= \Theta(1) \\ \frac{n^2}{p.(\frac{n^2}{p} + pn)} &= \Theta(1) \\ \frac{n^2}{n^2 + p^2 n} &= \Theta(1) \\ \Rightarrow p^2 &= O(n) \\ p &= O(\sqrt{n}) \end{aligned}$$

   $Thus$, the largest number of processors that can be used with optimal efficiency, $p_{max} = \Theta(\sqrt{n})$.

(b) Let M be the memory available per processor. If we scale number of processors p as $O(\sqrt{n})$, the memory M required per processor would scale as,

$$\begin{aligned} M &= \frac{n}{p} = \frac{n}{O(\sqrt{n})} \\ &= \Omega(\sqrt{n}) \end{aligned}$$

Therefore the required memory would also scale as a function of n in order to maintain the optimal efficiency. Because M is fixed, this cannot be asymptotically sustained. For larger values of n (i.e. $n = \Omega(M^2)$) we are forced to use $\Omega(\sqrt{n})$ processors and sacrifice the efficiency so as to be able to solve the problem at all. This points to another practical use of parallel computers – we are not able to solve the problem serially or using fewer processors at all, so this is the only alternative.

5. The maximum possible efficiency in this case is the optimal efficiency, therefore

$$E(p) = \frac{T(n,1)}{pT(n,p)} = \frac{\Theta(n^2)}{p.\Theta\left(\frac{n^2}{p} + \frac{n}{\sqrt{p}}\log p\right)} = \Theta(1)$$

After the simplification we get,

$$\begin{aligned} \sqrt{p}\log p &= O(n) \\ \Rightarrow 2\sqrt{p}\log\sqrt{p} &= O(n) \\ \Rightarrow \sqrt{p} &= O\left(\frac{n}{\log n}\right) \qquad \text{(Proof shown in class)} \\ \Rightarrow p &= O\left(\frac{n^2}{\log^2 n}\right) \end{aligned}$$