1. CSE 6220: HW1

Note: This assignment is submitted by: **Somdut Roy (GTID: sroy86)**
Colaborators were Daejin Kim and Han Gyol Kim (only for Q4).

1.1

The net effect of one pump and the leak = $(m-k)$ liters/sec.
The net effect of p pumps and the leak = $(pm-k)$ liters/sec.
$T(n,1) = \frac{n}{m-k}$ secs; $T(n,p) = \frac{n}{pm-k}$ secs.
$S(p) = \frac{T(n,1)}{T(n,p)} = \frac{pm-k}{m-k}$
Trivially for $p > 1, pm-k > pm-pk \Rightarrow pm-k > p(m-k) \Rightarrow \frac{pm-k}{m-k} > p \Rightarrow S(p) > p$.
Hence superlinear speedup has been achieved.
Theoretically, we are not supposed to get superlinear speedup but this case is a kind of exception where the problem size increases at a constant rate of k liters per second due to the leak. If the problem size was fixed or at least increasing in the order of p, we would not have seen a superlinear speedup.

1.2

$T(n,1) = n^2$
$T(n,p) = \frac{n^2}{p} + n$
Speedup $S(p) = \frac{T(n,1)}{T(n,p)} = \frac{n^2}{\frac{n^2}{p}+n} = \frac{np}{n+p}$
$\frac{dS(p)}{dp} = \frac{n^2}{(n+p)^2} > 0$ (always)
Hence $S(p)$ is a monotonously increasing function which implies that, **if we increase p, the speedup will keep on increasing (till it converges asymptotically to**

n).

For $n = kp$, $S(p) = \frac{p}{1+\frac{1}{k}}$ **which would mean that speedup will become independent of** n.

1.3

$A_1 : T(n,n^2) = \sqrt{n}$
$A_2 : T(n,n) = n$
Case 1: $p \leq n$.
$A_1 : T(n,p) = O(\frac{n^2 * \sqrt{n}}{p}) = O(\frac{n^{\frac{5}{2}}}{p})$.
$A_2 : T(n,p) = O(\frac{n*n}{p}) = O(\frac{n^2}{p})$.
Therefore, A_2 is faster than A_1 here.
Case 2: $p \geq n^2$.
$A_1 : T(n,p) = O(\sqrt{n})$.
$A_2 : T(n,p) = O(n)$.
Therefore, A_1 is faster than A_2 here.
Case 3: $n < p < n^2$.
There is a point inbetween where the two algorithm give similar runtime. We need to find that point.
$A_1 : T(n,p) = O(\frac{n^2 * \sqrt{n}}{p}) = O(\frac{n^{\frac{5}{2}}}{p})$.
$A_2 : T(n,p) = O(n)$.
So, $\frac{n^{\frac{5}{2}}}{p} = n \Rightarrow p = n^{\frac{3}{2}}$.
Therefore, for $p \leq n^{\frac{3}{2}}$, A_2 is faster whereas, for $p > n^{\frac{3}{2}}$, A_1 is faster.
Final result summary:
For $p \leq n^{\frac{3}{2}}$, A_2 is faster whereas, for $p > n^{\frac{3}{2}}$, A_1 is faster.

1.4

(a) $T(n,1) = n^2; T(n,p) = \frac{n^2}{p} + pn; E(p) = \frac{T(n,1)}{pT(n,p)} = \frac{1}{1+\frac{p^2}{n}}$. Therefore, for optimum efficiency, $p^2 \leq O(n) \Rightarrow p \leq O(\sqrt{n}) \Rightarrow p < c\sqrt{n}$ for some constant c.

(b) If the memory for each processor is M and at the maximum p for optimum efficiency, the memory used was $\frac{n}{p} = \frac{n}{c\sqrt{n}} = \frac{\sqrt{n}}{c}$, the number of processors can be scaled down by a factor of $f = \frac{c\sqrt{n}}{M}$ with efficiency remaining the same. In that case, the number of processors used will be $p_1 = fp = \frac{c\sqrt{n}}{M} * c\sqrt{n} = c^2 * n/M = O(n)$.Therefore we can scale the number of processors down to a certain extent.

1.5

$E(p) = \Theta(\frac{n^2}{p(\frac{n^2}{p} + \frac{nlogp}{\sqrt{p}})}) = \Theta(\frac{1}{1+\frac{\sqrt{p}logp}{n}})$.
$\sqrt{p}log.p \leq O(n)$. $\Rightarrow p \leq O((\frac{n}{logn})^2)$ (using trial and error).