CS 3101 Assignment 1

Alex Clarke 250592591

Problem 1

Question 1 :

a) $T(n) = O(nlog_2 n)$, for an array of size n.

b) Since a temporary array is created $S(n) = O(n)$.

Question 2 :

$n < \frac{Z}{L}$

Question 3 :

$Q(n) = \tau + 2Q(\frac{n}{2}) + \frac{\tau n}{Z}$, Where τ is the cost of a cache miss.

Question 4 :

$Q(n) = \tau + 2Q(\frac{n}{2}) + \frac{\tau n}{Z}$

$Q(n) = \tau + 2(\tau + 2Q(\frac{n}{2}) + \frac{\tau n}{Z}) + \frac{\tau n}{Z}$

$Q(n) = \tau(2^i + 1 + \frac{in}{Z})$, Where τ is the cost of a cache miss and i is s.t $2^i = n$.

$Q(n) = O(n)$

Question 5 :

Merge Sort is not an optimal sorting method for modern desktop/laptop computers since for large data sets the algorithim has high memory usage and high cache complexity, as a result negating any advantage gained from the $O(n \log_2 n)$ time complexity.

Problem 2

Question 1:

Access time: $100 + S - 1$

Locality: Temporal

Miss Type: Cold

Question 2:

Access time: $\frac{100S}{16} + \frac{15S}{16}$

Locality: Spacial

Miss Type: Cold

Question 3:

Access time: $100S$

Locality: None

Miss Type: Cold & Capacity

Question 4:

Access time: $200S$

Locality: None

Miss Type: Cold & Conflict

Problem 3

Question 1:

$Q(m,n) \approx \frac{2nmZ\tau}{L}$, where τ is the cost of a cache miss.

Question 2:

For large values of m,n there is a massive amount of capacity related cache misses which makes this a very slow algorithim, thus making it unsuitable for general use.

Question 3:

$Q(n,m) \approx \frac{Z}{L}((j-i)+(l-k)+(q-p))$ for $\alpha > \frac{j-i+1}{Z}$

Question 4:

I ran both methods for 2^{17} and found that the second method was about 30 percent faster. I have included the results in results.txt.