CS3334 Data Structures
(Suggested Solutions to Extra Exercises)

1. [Total 10 marks]

```
void Queue::enqueue(item x)           [3 marks]
{
   if (size==cap) realloc(3*cap);     [3 marks]
   arr[(front+size)%cap] == x;        [3 marks]
   size++;                            [1 mark]
}
```

2. [Total 20 marks]

   (a) [5 marks]
   The functions in non-decreasing order:

   $$\sqrt{n},\ n\log n,\ n^2\log n,\ n^3,\ 2^n$$

   (b) [5 marks]
   $f(n) = O(3n^2) = O(n^2)$

   (c) [10 marks]

   $$\begin{aligned} g(n) &= n + \frac{(n-1)(n)}{2} + 10 \\ &= n + \frac{n^2}{2} - \frac{n}{2} + 10 \\ &= O(n^2) \end{aligned}$$

   (Deduct 2 marks for each error/mistake in the derivation)

3. [Total 25 marks]

(a) [10 marks] Let P be the statement:

$$sum = \sum_{j=0}^{i-1} A[j]^2$$

3 marks: (Base case) When the program execution comes to the loop test for the first time, variable $i = 0$ and $sum = 0$. By the given notation, $\sum_{j=0}^{-1} A[j]^2 = 0$. That is, the statement P claims that $sum = 0$ which is true.

7 marks: (Induction step) Assume P is true when the program execution comes to the loop test for the k-th time for some $k \geq 1$. Then, $sum = \sum_{j=0}^{i-1} A[j]^2$ at that moment.

Suppose the loop test succeeds and the loop body is executed. After executing the statement "$sum+ = Array[i] * Array[i]$;", $sum = \sum_{j=0}^{i-1} A[j]^2 + A[i]^2 = \sum_{j=0}^{i} A[j]^2$. After executing the statement "i++;", $\sum_{j=0}^{i-1} A[j]^2$. Therefore, P is true again when the program execution comes to the loop test for the $(k+1)$-st time.

By the principle of mathematical induction, P is a loop invariant at the loop test.

(b) [5 marks]

3 marks: Proof of termination: Initially, $i = 0$. Each execution of the loop body increases i by 1. Eventually, i will become n and at that time, the loop test: $i < n$ will fail. So, the loop will terminate.

2 marks: Proof of total correctness: At the last loop test, i.e., when the loop test fails, $i = n$. By the loop invariant P, the variable sum contains the value $\sum_{j=0}^{n-1} A[j]^2$.

(c) [10 marks]

2 marks: The initialization of sum and i takes 2 units of time.

2 marks: The loop test is done $n+1$ times, costing $n+1$ units of time.

2 marks: The statement i++ is executed n times.

3 marks: The loop body is executed n times. Each execution of the loop body consists of one addition (+), one assignment (=), and one multiplication ($*$). Thus, each execution takes 3 units of time.

1 mark: Therefore, the worst case time complexity of the function is $T(n) = 2 + (n+1) + n + 3n = 5n + 3 = O(n)$.

4. [Total 10 marks]
   4 marks: Let $T(n)$ be the worst case time complexity for the function. The function perfoms at most one recursive call of size $n-1$. It also takes constant time to perform the local work. Therefore, $T(n) \leq T(n-1) + c$.

   3 marks: Solving the recurrence formula:

$$\begin{aligned} T(n) &\leq T(n-1) + c \\ T(n-1) &\leq T(n-2) + c \\ T(n-2) &\leq T(n-3) + c \\ &\cdots \\ T(1) &\leq T(0) + c. \end{aligned}$$

   Summing them up, we have

$$T(n) \leq T(0) + nc.$$

   where $T(0)$ is a constant.

   3 marks: Therefore, $T(n) = O(n)$.

5. [Total 15 marks]

   (a) [4 marks]
   Worst case time complexity of quicksort is $O(n^2)$.
   Average case time complexity of quicksort is $O(n \log n)$.

   (b) [6 marks]
   The function returns 0.

   (c) [5 marks]
   There are 4 inversions

——————— THE END ———————