Exam 01 Solutions

1. Ideally, to find the median, one needs to sort the array, then return either the average of the two middle values of the sorted array if the length of the array is even; otherwise, return the middle value of the sorted array. Hence, the pseudocode is as follows

```
Median(A)
1.  n <- A.length
2.  for i <- 0 to n - 1, do
  1.  m <- i
  2.  for j <- i + 1 to n - 1, do
    1.  if A[m] > A[j], then
      1.  m <- j
  3.  if m != i, then
    1.  t <- A[i]
    2.  A[i] <- A[m]
    3.  A[m] <- t
3.  if n % 2 = 0, then
  1.  return (A[n/2] + A[n/2-1]) / 2
4.  return A[n/2]
```

2. Given the pseudocode in question 1, the input size, n, is the size of the array input. Hence, the runtime table is

statement	time
1	1
2	$n+1$
2.1	n
2.2	$\sum_{i=0}^{n-1} n-i$
2.2.1	$\sum_{i=0}^{n-1} n-i-1$
2.2.1.1	$\sum_{i=0}^{n-1} n-i-1$
2.3	n
2.3.1	n
2.3.2	n
2.3.3	n
3	1
3.1	0
4	1

Thus, the runtime function, $T(n)$ is

$$\begin{aligned}
T(n) &= \sum_{i=1}^{13} c_i t_i \\
&= 6n + 4 + 3\sum_{i=0}^{n-1} n - 3\sum_{i=0}^{n-1} i - 2\sum_{i=0}^{n-1} 1 \\
&= 6n + 4 + 3n(n-1-0+1) - 3\frac{(n-1)(n)}{2} - 2(n-1-0+1) \\
&= 6n + 4 + 3n^2 - \frac{3}{2}n^2 + \frac{3}{2}n - 2n \\
&= \frac{3}{2}n^2 + \frac{11}{2}n + 4
\end{aligned}$$

3. The simulations of the algorithm only need to track the changes to the array input after each instance of the outer loop. Thus, we get

Simulation 1

```
1.  [3, 2, 8, 5, 1, 7, 4, 6]
2.  [1, 2, 8, 5, 3, 7, 4, 6]
3.  [1, 2, 3, 5, 8, 7, 4, 6]
4.  [1, 2, 3, 4, 8, 7, 5, 6]
5.  [1, 2, 3, 4, 5, 7, 8, 6]
6.  [1, 2, 3, 4, 5, 6, 8, 7]
7.  [1, 2, 3, 4, 5, 6, 7, 8]
8.  output = 4.5
```

Simulation 2

```
1.  [6, 3, 2, 9, 4, 5, 7, 1, 8]
2.  [1, 3, 2, 9, 4, 5, 7, 6, 8]
3.  [1, 2, 3, 9, 4, 5, 7, 6, 8]
4.  [1, 2, 3, 4, 9, 5, 7, 6, 8]
5.  [1, 2, 3, 4, 5, 9, 7, 6, 8]
6.  [1, 2, 3, 4, 5, 6, 7, 9, 8]
7.  [1, 2, 3, 4, 5, 6, 7, 8, 9]
8.  output = 5
```

4. Let $T(n) = \Theta(n^2)$. Then there exists constants c_1, c_2, and n_0 such that

$$c_1 n^2 \leq \frac{3}{2}n^2 + \frac{11}{2}n + 4 \leq c_2$$

whenever $n \geq n_0$. Letting $n_0 = 1$, implies

$$\begin{array}{ccccc}
c_1(1)^2 & \leq & \frac{3}{2}(1)^2 + \frac{11}{2}(1) + 4 & \leq & c_2(1)^2 \\
c_1 & \leq & \frac{3}{2} + \frac{11}{2} + 4 & \leq & c_2 \\
c_1 & \leq & 11 & \leq & c_2
\end{array}$$

and dividing by n^2 and letting n approaches ∞, yeilds

$$\begin{array}{ccccc}
c_1 & \leq & \frac{3}{2} + \frac{11}{2n} + \frac{4}{n^2} & \leq & c_2 \\
c_1 & \leq & \frac{3}{2} + 0 + 0 & \leq & c_2 \\
c_1 & \leq & \frac{3}{2} & \leq & c_2
\end{array}$$

Thus, letting $c_1 = \frac{3}{2}$, $c_2 = 11$, and $n_0 = 1$ satisfies the condition. Therefore, $T(n) = \Theta(n^2)$ as required.

5. The actual code is

```
#include <iostream>

double Median(double A[],int n)
{
  double t;
  for(int i = 0, m;i < n;i += 1)
  {
    m = i;
    for(int j = i + 1;j < n;j += 1)
    {
      if(A[m] > A[j]) {m = j;}
    }

    if(m != i)
    {
      t = A[i];
      A[i] = A[m];
      A[m] = t;
    }
  }
  if(n % 2 == 0) {return (A[n/2] + A[n/2-1]) / 2;}
  return A[n/2];
}

int main()
{
  double A[] = {3,2,8,5,1,7,4,6}, B[] = {6,3,2,9,4,5,7,1,8}, s = 0;

  std::cout << "The median for ( ";
  for(auto i :  A)
  {
    std::cout << i << "  ";
    s += i;
  }
  std::cout << ") is  "<< Median(A,8) << " and its average is "<< s / 8.0 << "\n";
  s = 0;
  std::cout << "The median for ( ";
  for(auto i :  B)
  {
    std::cout << i << "  ";
    s += i;
  }
  std::cout << ") is  "<< Median(B,9) << " and its average is "<< s / 9.0 << "\n";
}
```