1. True or False. [5 points each]

(a) **F** Computers are so fast today that complexity theory is only of theoretical, but not practical, interest.

(b) **T** There is a Las Vegas version of quicksort whose expected time complexity is $O(n \log n)$.

(c) **T** There is no comparison-based sorting algorithm for 5 items which uses fewer than 7 comparisons in the worst case.

(d) **T** There is a deterministic algorithm which finds the median of a set of $n$ numbers in $O(n)$ time.

2. Fill in the blanks. [5 points each blank.]

(a) What is the worst-case time complexity of binary search on a sorted array of size $n$? (Use $\Theta$ notation.)

   $O(\log n)$

(b) Heapsort is a sophisticated version of which one the following three simple algorithms – selection sort, insertion sort, bubblesort.

   **selection sort**

(c) Any comparison-based sorting algorithm for 5 items must use 7 comparisons in the worst case. (Exact answer, please.)

(d) Treesort is a sophisticated version of which one the following three simple algorithms – selection sort, insertion sort, bubblesort.

   **insertion sort**

(e) Name two well-known $\Theta(n \log n)$ time sorting algorithms.

   Any two of these three answers, or any other algorithm which satisfies the condition. **mergesort**, **heapsort**, **polyphase mergesort**.
3. Solve the recurrences. Give asymptotic answers in terms of \( n \), using either \( \Theta \), \( \Omega \), or \( O \), whichever is most appropriate. Use the master theorem or the anti-derivative method in each case. You may need to use substitution. [10 points each.]

(a) \( F(n) \geq 4F(n/2) + n^2 \quad F(n) = \Omega(n^2 \log n) \)
(b) \( G(n) = G(n - 1) + n^2 \quad G(n) = \Theta(n^3) \)
(c) \( H(n) = 2H(\sqrt{n}) + \log n \quad H(n) = \Theta(\log n \log \log n) \)
(d) \( J(n) = J(n - \sqrt{n}) + 1 \quad J(n) = \Theta(\sqrt{n}) \)
(e) \( K(n) \leq 2K(n/4) + \sqrt{n} \quad K(n) = O(\sqrt{n \log n}) \)
(f) \( L(n) = L(\log n) + 1 \quad L(n) = \Theta(\log^* n) \)
(g) \( G(n) \geq G(\sqrt{n}) + 1 \quad G(n) = \Omega(\log \log n) \)
(h) \( F(n) = F(n/2) + F(n - 1) + 1 \quad \text{???} \)

Of course there is a solution to (h), but I know of no way to express the solution in closed form. If anyone can come up with any insight on this problem, I’ll be grateful.

4. [15 points] Consider the following C++ code.

```cpp
void martha(int n)
{
    for(int i = 1; i < n; i = 2*n)
        cout << "Hello, George!" << endl;
}

void george(int n)
{
    for(int i = n; i > 1; i = i/2)
        martha(i);
}
```

What is the time complexity of `george(n)`?

\( \Theta(\log^2 n) \)

5. [30 points] Walk through polyphase mergesort, where the input file is as given below.

```
ABQXDFMGLKYT
ABQXDFMGLKYT
ABQX GLT
DFM KY T
ABDFMQX T
GKLTY
ABDFGKLMQTY
ABDFGKLMQTTY
```

```
6. [20 points] The following code implements an algorithm we’ve discussed in class, on an array A. What algorithm does the code implement?

```c
void swap(int&x,int&y)
{
    int temp = x;
    x = y;
    y = temp;
}

void main()
{
    for(int i = 0; i < n; i++)
        for(int j = i+1; j < n; j++)
}

selection sort
```

7. [20 points]

```c
int product(int a, int b)
{
    assert(b >= 0);
    int c = a;
    int d = b;
    int total = 0;
    while(b > 0)
    {
        if(d%2) total = total + c;
        c = 2*c;
        d = d/2;
    }
    return total;
}
```

(a) What does this function do?

(b) What is the loop invariant of the while loop?