1. Solve the following recurrence relation by guessing a tight bound.

   $T(n) = 2T(\lfloor n/2 \rfloor) + 2n\log_2 n$, $T(2) = 4$. (10pts)

2. Determine whether the following statements are correct or not. (2pts for each)

   (a) If a problem is *NP-complete*, then it can not be solved by any polynomial algorithm in worst cases.

   (b) If a problem is *NP-complete*, then we have not found any polynomial algorithm to solve it in worst cases.

   (c) If a problem is *NP-complete,* then it is unlikely that a polynomial algorithm can be found in the future to solve it in worst cases.

   (d) If a problem is *NP-complete*, then it is unlikely that we can find a polynomial algorithm to solve it in average cases.

   (e) If we can prove that the lower bound of an *NP-complete* problem is exponential, then we have proved that $NP \neq P$.

3. Given an array of integers A[1..n], such that, for all i, $1 \le i < n$, we have $|A[i] - A[i+1]| \le 1$. Let A[1]=x and A[n]=y, such that $x < y$. Design an efficient search algorithm to find j such that A[j] = z for a given value z, $x \le z \le y$. What is the maximal number of comparisons to z that your algorithm makes? (10pts)

4. Consider the following variation of Insertion Sort: For $2 \le i \le n$, to insert the key L[i] among $L[1] \le L[2] \le ... \le L[i-1]$, do a binary search to find the correct position for L[i].

   (a) How many key comparisons would be done in the worst case? (5pts)

   (b) What is the total number of times keys are moved in the worst case? (5pts)

5. Suppose the following algorithm is used to evaluate the polynomial

   $$p(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0.$$

   ```
   p:=$a_0$;
   xpower:=1;
   for i :=1 to n do
       xpower := x* xpower;
       p := p + $a_i$ * xpower;
   end
   ```

   (a) How many multiplications are done in the worst case? How many additions? (6pts)

   (b) How many multiplications are done on the average? (4pts)

6. Given one unsorted list (26, 5, 71, 1, 61, 11). (25pt)
   (a) Construct the max heap in linear time.
   (b) Implement the heap sort by (a).
   (c) Implement the quick sort.
   (d) Implement the merge sort.
   (e) Implement the nature merge sort.

7. Given the pattern : ABABABA, and string : AABABABCBABABABAB. (10pt)
   (a) Compute the failure function for the pattern.
   (b) Run the pattern matching by the failure function step by step.

8. Given the input chain F : 12→2→16→30→8→28→4→10→20→6→18, in range[0, 99], and radix = 10, #digit = 2. List the two passes of chains after using the following methods. (10pt)
   (a) MSD radix sort.
   (b) LSD radix sort.

9. Transfer the prefix form +/*A+BCDE into postfix form.(5pt)