CS 173, Spring 2015
Examlet 11, Part A

NETID:

FIRST:

LAST:

Discussion: Monday 9 10 11 12 1 2 3 4 5

```
01  maxpair(a_1, ..., a_n: an array of n positive integers, n ≥ 2)
02    if (n = 1) return 0
03    else if (n = 2) return a_1 + a_2
04    else
05        p = ⌊n/3⌋
06        q = ⌊2n/3⌋
07        rv = max(maxpair(a_1, ..., a_p), maxpair(a_{q+1}, ..., a_n))
08        for i=p to q
09            rv = max(rv, a_i + a_{i+1})
10        return rv
```

1. (5 points) Let $T(n)$ be the running time of maxpair. Give a recursive definition of $T(n)$.

2. (3 points) What is the height of the recursion tree for $T(n)$, assuming n is a power of 3?

3. (3 points) What is amount of work (aka sum of the values in the nodes) at level k of this tree?

4. (4 points) How many leaves does this recursion tree have? Simplify so that your answer is easy to compare to standard running times. Recall that $\log_b x = \log_a x \log_b a$.

CS 173, Spring 2015
Examlet 11, Part A

NETID:

FIRST:

LAST:

Discussion: Monday 9 10 11 12 1 2 3 4 5

```
1  procedure ComputeIt(a_1,..., a_n)  \\ input is an array of n positive integers
2      m := 0
3      for i := 1 to n − 1
4          for j := i + 1 to n
5              if |a_i − a_j| > m then m := |a_i − a_j|
6      return m
```

1. (4 points) What value does the algorithm return if the input list is $4, 13, 20, 5, 8, 10$

2. (4 points) Let $T(n)$ be the number of times that line 5 is executed. Express $T(n)$ using summation notation, directly following the structure of the code.

3. (4 points) Find an (exact) closed form for $T(n)$. Show your work.

4. (3 points) What is the big-theta running time of ComputeIt?

CS 173, Spring 2015
Examlet 11, Part A

NETID:

FIRST:

LAST:

Discussion: Monday 9 10 11 12 1 2 3 4 5

```
01 Magic(a_1, a_2, ... a_n: list of real numbers)
02   if (n = 1) then return 0
03   else if (n = 2) then return |a_1 − a_2|
04   else
05     L = Magic(a_2,a_3,...,a_n)
06     R = Magic(a_1,a_2,...,a_{n−1})
07     Q = |a_1 − a_n|
08     return max(L,R,Q)
```

Removing the first element of a list takes constant time; removing the last element takes $O(n)$ time.

1. (3 points) Give a succinct English description of what Magic computes.

2. (4 points) Suppose $T(n)$ is the running time of Magic. Give a recursive definition of $T(n)$.

3. (4 points) What is the height of the recursion tree for $T(n)$?

4. (4 points) How many leaves are in the recursion tree for $T(n)$?

CS 173, Spring 2015
Examlet 11, Part A

NETID:

FIRST:

LAST:

Discussion: Monday 9 10 11 12 1 2 3 4 5

```
01 Frog(p_1, ..., p_n : list of n 2D points, n ≥ 3)
02       if (n = 3)
03            return the largest of d(p_1, p_2), d(p_1, p_3), and d(p_2, p_3)
04       else
05            x = Frog(p_2, p_3, p_4, ..., p_n)      \\ removing p_1 from list takes constant time
06            y = Frog(p_1, p_3, p_4, ..., p_n)      \\ removing p_2 from list takes constant time
07            z = Frog(p_1, p_2, p_4, ..., p_n)      \\ removing p_3 from list takes constant time
08            return max(x, y,z)
```

The function $d(p, q)$ returns (in constant time) the straight-line distance between two points p and q.

1. (5 points) Suppose $T(n)$ is the running time of Frog on an input array of length n. Give a recursive definition of $T(n)$.

2. (4 points) What is the height of the recursion tree for $T(n)$?

3. (3 points) How many leaves are in the recursion tree for $T(n)$?

4. (3 points) What is the big-Theta running time of Frog?

CS 173, Spring 2015
Examlet 11, Part A

NETID:

FIRST:

LAST:

Discussion: Monday 9 10 11 12 1 2 3 4 5

```
00 diff(a_1, ..., a_n) : list of n positive integers, n ≥ 2)
01      if (n = 2) return |a_1 − a_2|
02      else
03          bestval = 0
04          for k = 1 to n
05              newval = diff(a_1, a_2, ..., a_{k−1}, a_{k+1}, ... a_n)   \\ constant time to remove a_k
06              if (newval > bestval) bestval = newval
07          return bestval
```

1. (3 points) Describe (in English) what diff computes.

2. (5 points) Suppose that $T(n)$ is the running time of diff on an input list of length n. Give a recursive definition of $T(n)$.

3. (3 points) What is the height of the recursion tree for $T(n)$?

4. (4 points) How many leaf nodes are there in the recursion tree for $T(n)$?

CS 173, Spring 2015
Examlet 11, Part A

NETID:

FIRST:

LAST:

Discussion: Monday 9 10 11 12 1 2 3 4 5

```
01 Getpower(k,n)   \\ inputs are natural numbers
02        if (n = 0) return 1
03        else if (n = 1) return k
04        else if (n is odd)
05              temp = getpower(k,floor(n/2))
06              return k*temp*temp
07        else
08              temp = getpower(k,floor(n/2))
09              return temp*temp
```

1. (5 points) Suppose $T(n)$ is the running time of Getpower. Give a recursive definition of $T(n)$.

2. (4 points) What is the height of the recursion tree for $T(n)$?

3. (3 points) How many leaves are in the recursion tree for $T(n)$?

4. (3 points) What is the big-Theta running time of Getpower?