CS 173, Spring 2016
Examlet 11, Part A

NETID:

FIRST:

LAST:

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

```
01 DoIt(a_1, ..., a_n)  \\ input is an array of n integers
02        if (n = 1) return a_1
03        else
04              m = ⌊n/2⌋
05              p = DoIt(a_1, ..., a_m)    \\ constant time to extract part of array
06              q = DoIt(a_{m+1}, ..., a_n)    \\ constant time to extract part of array
06              return max(p,q)
```

1. (5 points) Suppose that $T(n)$ is the running time of DoIt on an input array of length n and assume that n is a power of 2. Give a recursive definition of $T(n)$.

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

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

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

CS 173, Spring 2016
Examlet 11, Part A

NETID:

FIRST:

LAST:

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

```
01 Act(a_1, ..., a_n)  \\ input is a linked list of n integers
02       if (n = 1) return a_1
03       else
04            m = ⌊n/2⌋
05            p = Act(a_1, ..., a_m)    \\ O(n) time to split list
06            q = Act(a_{m+1}, ..., a_n)    \\ O(n) time to split list
06            return max(p,q)
```

1. (5 points) Suppose that $T(n)$ is the running time of Act on an input array of length n and assume that n is a power of 2. Give a recursive definition of $T(n)$.

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

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

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

CS 173, Spring 2016
Examlet 11, Part A

NETID:

FIRST:

LAST:

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

```
01 Weave(a_1, ..., a_n)  \\ input is a sorted array of n integers
02       if (n = 1) return a_1
03       else
04            m = ⌊n/2⌋
05            if a_m > 0
06                 return Weave(a_1, ..., a_m)    \\ constant time to extract part of array
07            else
08                 return Weave(a_{m+1}, ..., a_n)    \\ constant time to extract part of array
```

1. (5 points) Suppose that $T(n)$ is the running time of Weave on an input array of length n and assume that n is a power of 2. 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 does this tree have?

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

CS 173, Spring 2016
Examlet 11, Part A

NETID:

FIRST:

LAST:

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

```
01 Grind(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(Grind(a_1, ..., a_p), Grind(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 Grind. 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 2016
Examlet 11, Part A

NETID:

FIRST:

LAST:

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

```
01 Pump(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 = Pump(a_2,a_3,...,a_n)
06     R = Pump(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 Pump computes.

2. (4 points) Suppose $T(n)$ is the running time of Pump. 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 $T(n)$?