CS 473: Algorithms, Fall 2009
HBS 2

1. Consider the following variant of the selection algorithm that we saw in class[1].

```
ModifiedSelect(A, j):
    divide A into groups of 9 elements each
    find the median b_i of each group by brute force
    let B = {b_1, b_2, b_3, ..., b_⌈n/9⌉}
    b = ModifiedSelect(B, ⌈n/18⌉)
    partition A into A_less and A_greater using b as pivot
    if |A_less| = j
        return b
    else if |A_less| > j
        return ModifiedSelect(A_less, j)
    else
        return ModifiedSelect(A_greater, j − |A_less|)
```

   Analyze the running time of this modified algorithm by writing a recurrence relation for it and solving it. Briefly justify the recurrence relation that you derived.

2. (a) Euclid's algorithm for finding the greatest common divisor of two non-negative numbers a, b is the following.

```
Euclid(a, b):
    if b > a
        return Euclid(b, a)
    else
        if b = 0
            return a
        else
            return Euclid(b, a mod b)
```

   (i) Show by induction that the algorithm correctly computes the greatest common divisor of a and b.

   (ii) Assuming that the mod operation and other basic arithmetic operations take constant time, show that the running time of the algorithm is polynomial in the input size. [*Note that the input size is* $\Theta(\log a + \log b)$]

   (b) A slow version of the Euclid algorithm is the following.

```
SlowEuclid(a, b):
    if b > a
        return SlowEuclid(b, a)
    else
        if b = 0
            return a
        else
            return SlowEuclid(b, a − b)
```

[1] The selection algorithm that we saw in class divides A into groups of 5 elements.

Show via a class of examples that the above algorithm can take exponential time in the input size.