Assignment 2

Foundations of Algorithms

1. [20 pts, divide-and-conquer]

   Find and explain a divide-and-conquer solution to a problem (suitable) that we have not discussed in class or presented in the book. (For example, at a restaurant, friends searching an empty table in different parts of the dining area.)

2. [20 pts, recurrence]

   (a.) Show that the recurrence $T(n)=T(7n/10)+n$ has an upper bound complexity $T(n)=O(n)$

   (b.) Find and show an upper bound complexity for the recurrence $T(n)=2T(n/2)+n^4$

3. [20 pts, binary search]

   Given the binary (i.e. 2-ary) search algorithm as in the following, write the 4-ary search function. Empirically show that 4-ary search is faster with a sufficiently high input array size.

```
def bsearch(A, l, r, key):  # i.e. bsearch(A,0,len(A)-1,key)
    if l <= r:
        N_2 = (l+r)//2
        if A[N_2] == key:
            return N_2
        elif A[N_2] > key:
            return bsearch(A, l, N_2-1, key)
        else:
            return bsearch(A, N_2+1, r, key)
    else:
        return -1
```

4. [40 pts, divide-and-conquer]

   Solve problem 4-5 (page 109), chip testing problem.