HW week2

Name: Yuan Li

Email: [liyu4@oregonstate.edu](mailto:liyu4@oregonstate.edu)

**Problem 1:**

a. T(n) = T(n-1) + 2n

= T(n-2) + 2(n-1) + 2n = T(n-2) + 4n – 2

= T(n-3) + 2(n-2) + 2n + 2n -2 = T(n-3) + 6n -6

…

= T(1) + 2n(n-1) – 2(n-2)

= 1+ 2n^2 – 2n – 2n + 4

= c + 2n^2 -4n

= θ(n^2)

b. T(n) = T(n-2) +3

= T(n-4) + 3 +3

= T(n-6) + 3 +3 +3

…

= T(1) + ((n-2)/2 )\* 3

= 1 + 3(n-2)/2

=θ(n)

c. T(n) = 4T(n/2) + 3n^2

Mater Method: a =4, b = 2

n^logba = n^log24 = n^2

case 2: f(n) = n^logba = θ(n^2)

d. T(n) = 2T(n/4) + n^2

Master Method: a = 2, b = 4

n^logba = n^log42 = n^(1/2)

f(n) = n^2= Ω(n^logba)

af(n/b) = 2(n/4)^2 = n^2/8< cf(n), for c = 0.5

Therefore, T(n) = θ(n^2)

**Problem 2:**

a. The quaternary search algorithm searches for a value in a list. First call is

quaternary (A, first, last, target).

quarter1 = (last-first) / 4 + first,

quarter2 = (last-first) /2 + first,

quarter3 = (last-first) /4 \* 3 + first,

if any of quarter1, quarter2, quarter3 matches target, return the index.

if A[first]≤target≤A[quarter1]

call quaternary (A, first, quarter1, target)

if A[quarter1]<target<A[quarter2]

call quaternary (A,quarter1, quater2,target)

if A[quarter2]<target<A[quarter3]

call quaternary (A,quarter2, quater3,target)

if A[quarter3]<target<A[quarter4]

call quaternary (A,quarter3, last, target)

b .T(n) = T(n/4) + c

c. Master Method: a = 1, b = 4

n^log41 = 1

f(n) = θ(n^logba)

case 1: T(n) = θ (lgn)

Problem 3:

T(n) = 3T(2n/3) + c

Master Method: a =3, b= 3/2

n^logba = n^log3/23

2.70<log3/23 <2.71

So, n^logba > n^2.7

f(n) = O(n^logba-)

T(n) = θ(n^ log3/23)