# CSCI 665 Assignment 3

2.Consider a = [10, 20, 30, 40, 50] ; v = 50

search([10, 20, 30, 40, 50]; 50)
= searchHelp([10, 20, 30, 40, 50], 50; 0, 4)
= searchHelp([10, 20, 30, 40, 50], 50; 2,4)
= searchHelp([10, 20, 30, 40, 50], 50; 2,3)
= searchHelp([10, 20, 30, 40, 50], 50; 2,3)
= searchHelp([10, 20, 30, 40, 50], 50; 2,3)
= searchHelp([10, 20, 30, 40, 50], 50; 2,3)
= searchHelp([10, 20, 30, 40, 50], 50; 2,3)
Due to the incorrect Algorithm for binary search,
Here it is stuck on an infinite loop.

3.a.
$S1 = B12 - B22 = 8-2=6$
$S2 = A11 + A12 = 1+3=4$
$S3 = A21 + A22 = 7+5=12$
$S4 = B21 - B11 = 4-6=-2$
$S5 = A11 + A22 = 1+5=6$
$S6 = B11 + B22 = 6+2=8$
$S7 = A12 - A22 = 3-5=-2$
$S8 = B21 + B22 = 4+2=6$
$S9 = A11 - A21 = 1-7=-6$
$S10 = B11 + B12 = 6+8=14$
The products are:
$P1 = A11 * S1 = 1*6=6$  $P2 = S2 * B22 = 4*2=8$  $P3 = S3 * B11 = 12*6=72$
$P4 = A22 * S4 = 5*-2=-10$
$P5 = S5 * S6 = 6*8=48$
$P6 = S7 * S8 = -2*6=-12$
$P7 = S9 * S10 = -6*14=-84$
The four matrices are:
$C11 = P5 + P4 - P2 + P6 = 48+(-10)-8+(-12)=18$
$C12 = P1 + P2 = 6+8=14$
$C21 = P3 + P4 = 72+(-10)=62$
$C22 = P5 + P1 - P3 - P7 = 48+6-72-(-84)=66$

Hence the resultant Matrix is:-[[18,14],
[62,66]]

b.Strassens Algorithm
MatMul(A,B)

n = A.rows
If n == 1
C11 ← A11 * B11 Else
S1 ← B12 - B22
S2 ← A11 + A12
S3 ← A21 + A22
S4 ← B21 - B11
S5 ← A11 + A22
S6 ← B11 + B22
S7 ← A12 - A22
S8 ← B21 + B22
S9 ← A11 - A21
S10 ← B11 + B12

P1 ← MatMul (A11,S1) P2 ← MatMul (S2,B22) P3 ← MatMul (S3,B11) P4 ← MatMul (A22,S4) P5 ← MatMul (S5,S6)
P6 ← MatMul (S7,S8)
P7 ← MatMul (S9,S10)

C11 ←P5 +P4 −P2 +P6 C12 ← P1 + P2 C21 ← P3 + P4 C22 ←P1 +P5 −P3 −P7
Return C

c.
Given: C21 = P3 + P4
We know that,

P3 = S3 * B11
= (A21 + A22) * B11
P4 = A22 * S4
= A22 (B21 - B11)
P3 + P4
= (A21 + A22) * B11 + A22 (B21 - B11)
= A21* B11 + A22 * B11 + A22 * B21 - A22 * B11
= A21* B11 + A22 * B21
Hence proved.

(d)
Given: C22 = P5 + P1 - P3 - P7
We know that,
P5 = S5 * S6
= (A11 + A22) (B11 + B22)
= A11 * B11 + A11 * B22 + A22 * B11 + A22 * B22
P1 = A11 * S1

= A11 (B12 - B22)
= A11 * B12 - A11 * B22
P3 = S3 * B11
= (A21 + A22) B11
= A21 * B11 + A22 * B11
P7 = S9 * S10
= (A11 - A21) (B11 + B12)
= A11 * B11 + A11 * B12 - A21 * B11 - A21 * B12
P5 + P1 = A11 * B11 + A11 * B22 + A22 * B11 + A22 * B22 + A11 * B12 - A11 * B22
= A11 * B11 + A22 * B11 + A22 * B22 + A11 * B12
P5 + P1 - P3 = A11 * B11 + A22 * B11 + A22 * B22 + A11 * B12 - (A21 * B11 + A22 * B11)
= A11 * B11 + A22 * B22 + A11 * B12 - A21 * B11
P5 + P1 - P3 - P7
= A11 * B11 + A22 * B22 + A11 * B12 - A21 * B11 - (A11 * B11 + A11 * B12 - A21 * B11 - A21 * B12)
= A22 * B22 + A21 * B12
Hence proved.

e.
T(1) = 1

T(n) = 7 T(n/2) + 9/2 n^2

= 7 (7 T(n/2^2) + 9/2 (n/2)^2) + 9/2 n^2

= 7^2 T(n/2^2) + 7.9/2 (n/2)^2 + 9/2 n^2

= 7^2 (7 T(n/2^3) + 9/2 (n/2^2)^2) + 7.9/2 (n/2)^2 + 9/2 n^2

= 7^3 T(n/2^3) + 7^2 9/2 (n/2^2)^2) + 7.9/2 (n/2)^2 + 9/2 n^2

Identifying the pattern

= 7^k T(n/2^k) + 7^(k-1) 9/2 (n/2^(k-1))^2) + ... + 7^0 9/2 (n/2^0)^2

= 7^k T(2^m/2^k) + 7^(k-1) 9/2 (2^m/2^(k-1))^2) + ... + 7^0 9/2 (2^m/2^0)^2

= 7^k T(2^{m-k}) + 7^(k-1) 9/2 (2^{m-(k-1)})^2) + ... + 7^0 9/2 (2^{m-0})^2 consider k = m

T(2^m) = 7^m T(2^{m-m}) + 7^(k-1) 9/2 (2^{m-(m-1)})^2) + ... + 7^0 9/2 (2^{m-0})^2

= 7^m T(1) + 7^(k-1) 9/2 (2^{1})^2) + ... + 7^0 9/2 (2^{m})^2

= 7^m + 9/2 sum_{i=0}^{m-1} 7^i (2^{m-1})^2

= 7^m + 9/2 sum_{i=0}^{m-1} 7^i (2^{2m-2})

= 7^m + 9/2 2^m sum_{i=0}^{m-1} (7 / (2^{2}))^i\\

= 7^m + 9/2 2^m ((7 / (2^{2}))^m - 1)/(7/2^2)-1\\

= 7^m + 9/2 2^m ((7^m/2^{2m}) - 1)

= 7^m + 9/2 2^m (7^m - 2^{2m})/2^{2m}

= 7^m + 6 (7^m/2^m) - 2^m

T(n) = 7^m + 6 (7^m/n) - n

f.
Let m be the smallest power of 2 which is greater than n.
Then,
Adding (m-n) zeros to the nxn matrices to make them mxm matrices and perform multiplication using Strassen's algorithm.
Then the resulting algorithm will run at time. $\Theta(\text{n lg}(7))$
Hence Proved.

Suppose 2^(k- 1) < n < m = 2^k.
Now, we could extend original n ∗ n matrices to m∗ m matrices by appending zeros for the Strassen's algorithm to work on extended matrices.
Then, to eliminate the appended elements from the matrices needs O(n^2).
Because 2^(k-1) < n, it follows that m < 2n. Therefore, the runtime becomes
Θ((2n)^lg 7) = Θ(2^(lg 7)n^(lg 7)) = Θ(n^(lg 7))

(g)
The three multiplications are
ac, bd and (a + b) (c + d)
Computing,
ac - bd + i((a+b)(c+d) - ac - bd)
ac - bd + i(ac + ad + bc + bd - ac - bd)
= ac - bd + i(ad + bc)

Hence the real component is ac - bd and imaginary component is ad + bc

4.a.
For every even n ∈ N, there exist some k ∈ N such that n = 2k.
Then, [(n + 1)/2] = [(2k + 1)/2]
= [k + 1/2]
= k
= [k]
= [2k/2]
= [n/2] , because n = 2k
Same way for every odd n ∈ N, there exist some k ∈ N such that n = 2k + 1.
Then, [(n + 1)/2] = [((2k + 1) + 1)/2]
= [(2k + 2)/2]
= [k + 1]
= k + 1
= [k + 1/2]
= [(2k + 1)/2]
= [n/2] , because n = 2k + 1
So, for any n ∈ N, [(n + 1)/2] = ceil of (n/2)

b.
For every even n ∈ N, there exist some k ∈ N such that n = 2k.
Then, [n/2] + 1 = [2k/2] + 1
= [2k/2] + 1
= [k] + 1
= k + 1
= [k + ½]
= [(2k + 1)/2]
= [(n + 1)/2] , because n = 2k
Same way for every odd n ∈ N, there exist some k ∈ N such that n = 2k + 1.
Then, [n/2] + 1
= [(2k + 1)/2] + 1
= [k + ½] + 1
= [k] + 1
= k + 1
= [k + 1]
= [2(k + 1)/2]
= [(2k + 2)/2]
= [((2k + 1) + 1)/2]
= [(n + 1)/2] , because n = 2k + 1
So, for any n ∈ N, [n/2] + 1 = ceil of (n + 1)/2

c.
Given D(n) = T(n + 1) ) T(n)

So, $D(1) = T(1 + 1) ) T(1)$
$= T(2) ) T(1)$
$= T(2) ) 0 = T(2)$
$= T([2/2]) + T([2/2]) + 2$ , From recurrence definition
$= T([1]) + T([1]) + 2$
$= T(1) + T(1) + 2$
$= 0 + 0 + 2$
$= 2$
Now, $D(n) = T(n + 1) )$
$T(n) = T([(n + 1)/2])+T([(n + 1)/2])+(n+1)) (T([n/2])+T([n/2])+n)$
, by recurrence definition
$= T([(n + 1)/2]) + 1 - T([n/2]) )$
, putting value from 4.a.
$= T([n/2] + 1) + 1 = T([n/2])$ , putting value from 4.b
$= T([n/2] + 1) )- T([n/2]) + 1$
$= D([n/2]) + 1$ , by definition of $D(n)$

d.
By the strong form of mathematical induction.
Observe $D(1) = 2$ , from 3.c
$= 0 + 2 = [\lg 1] + 2$
Assume $D(k) = [\lg(k)] + 2$ if $0 < k < n$
Now,
$D(n) = D([n/2]) + 1$, from 3.c
Suppose, $[n/2] = k$ for any $k \in N$ and $k \geq 1$
So, $D(n) = D([k]) + 1$
$= D([k]) + 1$
$= [\lg k] + 2 + 1$ , by assumption
$= [\lg(n/2)] + 2 + 1$ , because $k = n/2$
$= [\lg n - \lg 2] + 2 + 1$
$= [\lg n – 1] + 2 + 1$
$= [\lg n] – [1] + 2 + 1$ , for any $n \in N$, $[\lg n – 1]$
$= [\lg n] – [1]$
$= [\lg n] + 2$

e.
$T(n) -T(1) = T(n) = T(n n 1) + T(n n 1) - T(n n 2) + T(n n 2) − \cdots + T(3)- T(2) + T(2) - T(1)$
$= D(n - 1) + D(n - 2) + \cdots + D(2) + D(1)$ , from definition of $D(n)$
$= D(1) + D(2) + \cdots + D(n n-2) + D(n - 1)$
$= n-1\Sigma k=1 D(k)$
Hence, $T(n) - T(1) = -1\Sigma k=1 D(k)$
Here, $T(1) = 0$, and from 4.d. we conclude our immediate consequence is
$T(n) = n-1\Sigma k=1 [\lg k] + 2$

f.

$T(n) = {}^{n-1}\Sigma_{k=1}$ [lg k] + 2

= $^{n}\Sigma_{k=1}$ [lg k] + 2 - ([lg n] + 2) , adding n th term to change the limit

= (([lg 1] + 2) + ([lg 2] + 2) + · · · + ([lg n] + 2)) - ([lg n] + 2)

= (([lg 1] + [lg 2] + · · · + [lg n]) + 2n) - ([lg n] + 2)

= (([lg 1 + lg 2 + · · · + lg n]) + 2n) - ([lg n] + 2) , for arbitrary larger n combining all sums of log into a single floor.

= ([lg(n!)] + 2n) - ([lg n] + 2)

= [lg(n!)] – [lg n] + 2n - 2

= [lg(n!)] – [lg n] + 2(n - 1)

Time complexity of above equation in terms of Big-O = O(n lg(n)) − O(lg n) + O(n)

So, overall time complexity of T(n) = O(n log(n)).