CSE 321
HW 2

1)

a) $T(n) = 3T(n-1) - 2T(n-2)$ $T(1)=1$ $T(2)=2$

$$\frac{r^n}{r^{n-2}} = \frac{3r^{n-1} - 2r^{n-2}}{r^{n-2}}$$

$r^2 = 3r - 2$

$r^2 - 3r + 2 = 0$

$r = 2$

$r = 1$

$T(n) = c_1 1^n + c_2 2^n$

$T(1) = c_1 + 2c_2 = 1$

$T(2) = c_1 + 4c_2 = 2$

$c_2 = \frac{1}{2}$ $c_1 = 0$

$$\boxed{T(n) = \frac{1}{2} 2^n \quad \Theta(2^n)}$$

b) $T(n) = T(n/2) + 1$ $T(1)=1$ $T(2)=2$

$n = 2^k$ $k = \log_2 n$

$$\left.\begin{aligned} T(2^k) &= T(\cancel{2^{k-1}}) + 1 \\ \cancel{T(2^{k-1})} &= \cancel{T(2^{k-2})} + 1 \\ &\vdots \\ k=1 \quad \cancel{T(2)} &= T(1) + 1 \end{aligned}\right\} k$$

$+$

$T(2^k) = T(1) + k$

$$\boxed{T(n) = 1 + \log_2 n \quad \Theta(\log n)}$$

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

$T(n) = aT(\frac{n}{b}) + f(n)$

$f(n) \in \Theta(n^d)$

$T(n) \in \begin{cases} \Theta(n^d) & \text{if } a < b^d \\ \Theta(n^d \log n) & \text{if } a = b^d \\ \Theta(n^{\log_b a}) & \text{if } a > b^d \end{cases}$ → Masters Theorim

$a = 4, \quad b = 2, \quad f(n) = n^2$

$f(n) \in \Theta(n^2) \quad d = 2$

if $a = b^d \quad \Theta(n^d \log n)$

$4 = 2^2$

$\boxed{T(n) \in \Theta(n^2 \log n)}$

e) $T(n) = 2T(n/2) + O(n)$

$a = 2 \quad b = 2 \quad d = 1$

if $a = 2^1$ ✓

$\boxed{T(n) = \Theta(n . \log n)}$

g) $T(n) = T(n/2) + n$

$a = 1 \quad b = 2 \quad f(n) \in O(n)$

$a < b^d \quad d = 1$

$1 < 2^1$

$\boxed{T(n) \in \Theta(n)}$

h) $T(n) = 2T(\sqrt{n}) + 1$ $\quad T(1)=1 \quad T(4)=3$

$n = 2^{2^k} \quad k = \log_2(\log_2 n)$

$T(2^{2^k}) = 2T(2^{\frac{2^k}{2}}) + 1$

$T(2^{2^k}) = 2T(2^{2^{k-1}}) + 1$

$X(k) = T(2^{2^k})$

$X(k) = 2X(k-1) + 1$

$X(k) = X_H(k) + X_p(k)$

$\underline{X_H(k)}$

$r - 2 = 0$

$r = 2 \quad X_H(k) = c\,2^k$

$X_p(k) = D$ $\quad T(2^{2^k}) = 2 \cdot 2^k + D$

$X(k) = c\,2^k + D$

$T(2^{2^k}) = c\,2^k + D$

$T(1) = c + D = 1$

$T(4) = 2c + D = 3$

$c = 2 \quad D = -1$

$$\boxed{\begin{array}{c} T(n) = 2\log_2 n - 1 \\ T(n) \in \Theta(\log n) \end{array}}$$

2) Procedure abs (x)

if $x < 0$ then

return $(-1) * x$

endif

return x

endprocedure

Procedure max (x, y)

if $x > y$ then

return x

endif

return y

endprocedure

Procedure height_of_tree (root)

if root equals null then

return 0

endif

return 1 + max (height_of_tree (root.left), height_of_tree (root.right))

endprocedure

$T(n) = 2T(\frac{n}{2}) + 1$

Masters Theorem $\rightarrow$ $a = 2$ $b = 2$ $d = 0$ $a > b^d$ $T(n) \in \Theta(n^{\log_b a})$

$2 > 1$ $T(n) \in \Theta(n)$

Procedure is_balanced (root)

if root == null than

return true

endif

lh = height (root.left)

rh = height (root.right)

return abs (lh - rh) <= 1 && is balanced (root.left), && is balanced (root.right)

endprocedure

$T(n) = 2T(\frac{n}{2}) + 2O(n)$ $a = b^d$, $2 = 2$, $T(n) \in \Theta(n \log n)$

$a = 2$ $b = 2$ $d = 1$

3) a) $T(n) = 5T(n/2) + O(n^3)$

Masters Theorem

$T(n) = a\,t(\frac{n}{b}) + f(n)$

$f(n) \in \Theta(n^d)$

$$T(n) \in \begin{cases} O(n^d) & \text{if } d > \log_b a \\ O(n^d \log n) & \text{if } d = \log_b a \\ O(n^{\log_b a}) & \text{if } d < \log_b a \end{cases}$$

$a = 5 \quad b = 2 \quad d = 3$

$3 > \log_2 3$

$T(n) \in O(n^3)$

b) $T(n) = 2T(n-2) + O(n)$

$k = n/2 \quad n = 2k$

$T(2k) = 2T(2k-2) + O(2k)$

$X(k) = T(2k)$

$X(k) = 2X(k-1) + O(2k)$

$a = 2 \quad b = 1 \quad d = 1$

$4 < 2^2$

$X(k) \in O(2k)$

$T(n) \in O(n)$

c) $T(n) = 3T(n/2) + O(n^2)$

$a = 3 \quad b = 2 \quad d = 2$

$2 > \log_2 3$

$T(n) \in O(n^2)$

- Algorithm B is a recursive algorithm with linear time complexity making it efficient for bigger input sizes.

5) foo(n): $\rightarrow T(n)$

```
if n <= 1:
    return 1
else:
    for i in range(n):
        print("a")
return foo(n/2) + foo(n/2)
```

$\rightarrow O(1)$

$\rightarrow O(n)$

$\rightarrow 2T(n/2)$

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n)$$

<u>Masters Theorem</u>

$a = 2 \quad b = 2 \quad d = 1$

$a = b^d \qquad T(n) \in \Theta(n^d \log n)$

$2 = 2^1 \qquad \boxed{T(n) \in \Theta(n \log n)}$