CSE 321 - Homework 3

Harun ALBAYRAK - 171044014

1)

a) $T(n) = 27T(n/3) + n^2 \Rightarrow a = 27,\ b = 3,\ f(n) = n^2$

$n^{\log_b a} = n^{\log_3 27} = n^{\log_3 3^3} = n^3$

Since $n^3 > f(n)$, $T(n) = \Theta(n^{\log_b a})$ for $O(n^{\log_b a - \varepsilon})$

$\boxed{T(n) = \Theta(n^3)}$

b) $T(n) = 9T(n/4) + n \Rightarrow a = 9,\ b = 4,\ f(n) = n$

$n^{\log_b a} = n^{\log_4 9} = n^{1,58}$

Since $n^{1,58} > f(n)$, $T(n) = \Theta(n^{\log_b a})$

$\boxed{T(n) = \Theta(n^{1,58})}$

c) $T(n) = 2T(n/4) + \sqrt{n} \Rightarrow a = 2,\ b = 4,\ f(n) = \sqrt{n}$

$n^{\log_b a} = n^{\log_4 2} = n^{0,5} = n^{1/2} = \sqrt{n}$

Since $\sqrt{n} = f(n)$, $T(n) = (n^{\log_b a} . \log n)$

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

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

$n = 2^m \Rightarrow T(2^m) = 2T(2^{m/2}) + 1$

$S(m) = 2T(m/2) + 1 \Rightarrow$ master theorem $(a = 2, b = 2, f(n) = 1)$

$m^{\log_b a} = m^{\log_2 2} = m > f(n)$, so $S(m) = T(2^m) = \Theta(m)$

Since $T(2^m) = \Theta(m) \Rightarrow \boxed{T(n) = \Theta(\log n)}$

e) $T(n) = 2T(n-2)$

$T(n) = 2[2T(n-4)] \Rightarrow 2^2 T(n-2-2)$

$T(n) = 2^3 T(n-2-2-2)$

$T(n) = 2^k T(n-2k)$ Assume $\Rightarrow n - 2k = 0 \Rightarrow n = 2k \Rightarrow k = n/2$

$T(n) = 2^{n/2} . T(0) \Rightarrow T(0) = 1 \Rightarrow T(n) = 2^{n/2}$

$\boxed{T(n) = \Theta(2^{n/2})}$

f) $T(n) = 4T(n/2) + n$, $T(1) = 1$, $a = 4$, $b = 2$, $f(n) = n$

$n^{\log_b a} = n^{\log_2 2^2} = n^2$

Since $n^2 > f(n)$, $\boxed{T(n) = \Theta(n^2)}$

---

g) $T(n) = 2T(\sqrt[3]{n}) + 1$

$n = 3^m \implies T(3^m) = 2T(3^{m/3}) + 1$

$S(m) = 2T(m/3) + 1 \implies$ Master theorem ($a=2$, $b=3$, $f(n) = 1$)

$m^{\log_b a} = m^{\log_3 2} = m^{0,63} > 1$, so $S(m) = T(3^m) = \Theta(m^{0,63})$

$\boxed{T(n) = \Theta(\log_3 n^{0,63})}$

---

2) sub problem size $= n/2$

sub problem count = for loop (1 to n) $= 2^k$ power of 2

print_line ("**") = 1

$T(n) = 2^k T(n/2) + 1$, $a = 2^k$, $b = 2$, $f(n) = 1$

$n^{\log_b a} = n^{\log_2 2^k} = n^k$

Since $n^k > f(n)$, $T(n) = \Theta(n^k)$

---

3) sub problem size $= 2n/3$

Sub problem count $= 3$

Other if $= 1$

$T(n) = 3T(2n/3) + 1 \implies T(n) = 3T(n/(3/2)) + 1$

Master theorem $\implies a = 3$, $b = 3/2$, $f(n) = 1$

$n^{\log_b a} = n^{\log_{1,5} 3} = n^{2,71}$

Since $n^{2,71} > 1$, $T(n) = \Theta(n^{2,71})$

4) Input array = [10, 7, 8, 9, 1, 5, 2, 4, 3, 1]

I have implemented quick sort and insertion sort with counting swap number. (question4.py)

The number of swap operations (Insertion sort) : 35

The number of swap operations (Quick sort) : 13

Also, Quicksort average-case complexity is $O(n \log n)$ because quicksort is recursive sorting algorithm.

On the other part, Insertion sort average-case complexity is $O(n^2)$ because insertion sort contains 2 nested loops.

New input array = [-5, -3, 4, 2, 25, 67, 0, -2, 6, 42, 10, 3, -7, 3, 6, 7, 1]

The number of swap op. (Insertion sort) : 59

The number of swap op. (Quick sort) : 44

Eventually, As can be seen from the above data and comparisons of their average-case complexities, quicksort is better algorithm than insertion sort.

5)

a) $T(n) = 5T(n/3) + O(n^2) \Rightarrow$ Master theorem ($a=5$, $b=3$, $d=2$)

$\log_b a = \log_3 5 = 1{,}46$, $1{,}46 < 2$ ($d > \log_b a$)

Since $d > \log_b a$, $\boxed{T(n) = O(n^2)}$

b) $T(n) = 2T(n/2) + n^2 \Rightarrow$ Master theorem ($a=2$, $b=2$, $d=2$)

$\log_b a = \log_2 2 = 1$, $1 < 2$ ($d > \log_b a$)

Since $d > \log_b a$, $\boxed{T(n) = O(n^2)}$

c) $T(n) = T(n-1) + n$

$T(n) = [T(n-2) + n-1] + n$

$= T(n-2) + (n-1) + n$

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

$T(n) = T(\underbrace{n-k}) + (n-(k-1)) + (n-(k-2)) + \ldots\ldots + (n-1) + n$

Assume $n-k=0$

$\therefore\ n=k$

$T(n) = T(0) + 1+2+3+ \ldots\ldots + (n-1) + n$

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

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