Hw1

Problem 3

Part (a) $T(n) = \overset{\text{int } i=2}{\Theta(1)} + \overset{\text{while loop}}{\sum_{i=2}^{n-1}} (\Theta(1))$

first iteration $i = 4 = 2^2$

second iteration $i = 16 = 2^4$

$\vdots$

k^{th} iteration $i = 2^{2k}$

$2^{2k} = n$

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

$\therefore T(n) = \Theta(1) + \Theta(\log(n)) \approx \Theta(\log(n))$

Part (b) $T(n) = \underset{\text{Outer For loop}}{\sum_{i=1}^{n-1}} \left(\Theta(1) + \Theta\left(\underset{\text{inner For loop Assuming if is always True}}{\sum_{k=0}^{i^3}} \Theta(1) \right) \right)$

if statement : $i \% \sqrt{n} == 0$

$n = 1$ $i = 1,$

$n = 2$ $i = 1, 2$

$n = 4$ $i = 2, 4,$

$\vdots$

n $i = \sqrt{n}, 2\sqrt{n}, \dots n$

# of iterations $= \frac{n}{\sqrt{n}} = \sqrt{n}$

$\therefore \Theta\left(\sum_{k=0}^{i^3} \Theta(1)\right) \Rightarrow \sum_{c=1}^{\sqrt{n}} \sum_{k=0}^{i^3} \Theta(1) = \sum_{c=1}^{\sqrt{n}} \Theta(i^3) = \Theta(\sqrt{n}\, i^3)$

$\because i = c\sqrt{n}$

$\Theta(\sqrt{n}\, i^3) = \Theta(n^{\frac{1}{2}} \cdot n^{\frac{1}{2}}) = \Theta(n)$

$\therefore T(n) = (n-1)\Theta(n)$

$= \Theta(n^2)$

outer for loop $\Downarrow$ for(int k=1; k<=n; k++) $\Downarrow$ inner for loop $\Downarrow$

Part(c) $T(n) = \sum_{i=1}^{n} \left(\Theta(1) + \sum_{k=1}^{n} \left(O\left(\sum \Theta(1) + \Theta(1) \right) \right) \right)$

$\Uparrow$ Assume if(A[k]==i) is always True

since m doubles until m>n in the inner loop $\therefore$ for last iteration $2^k = n$, $k = \log_2(n)$

$$O\left(\sum \Theta(1)+\Theta(1) \right) = O\left(\sum_{k=1}^{\log_2(n)} \Theta(1) \right) = O(\log(n))$$

Since for any Array A[], every index k can only point to one value i and every value i can point to one or more than one indicies, making the function described to be injective

$\therefore$ The if statement if(A[k]==i) is True n times for a set of (k,i) s.t. A[k]==i as there is only n set of (k,i) that satisfies A[k]==i

$$\therefore T(n) = \sum_{i=1}^{n} \left(\sum_{k=1}^{n} (O(\log n)) \right) = \Theta(n\log n)$$

a[i]=i*i; $\downarrow$ inner for loop $\downarrow$

Part(d) $T(n) = \sum_{i=0}^{n-1} \left(\Theta(1) + O\left(\Theta(1) + \sum_{j=0}^{size} \Theta(1) \right) \right)$

$\uparrow$ outer for loop $\uparrow$ Assume if(i==size) is always True

For 1st Resize	size = 10	newsize = 15
2nd	size = 15	newsize = 22
3rd	size = 22	newsize = 33
4th	size = 33	newsize = 49
	⋮	
k^{th}	size = $10 \times 1.5^{k-1}$	new size = 10×1.5^{k}

$$k-1 = \log_{1.5}\left(\frac{size}{10}\right) \approx \log\left(\frac{size}{10}\right) \Rightarrow \log(n)$$

$\therefore$ For an integer $n \geq 10$, $\log(n)$ times of resizes are needed

$$\therefore T(n) = \sum_{i=0}^{n-1} (\Theta(1) + \log(n)) = \Theta(n\log n)$$

Problem 4

Question A:

in1 = 1 → 2 → 3 → 4   in2 = 5 → 6

1 → next = llrec (5 → 6, 2 → 3 → 4)   return $\boxed{\text{in1} = 1 \to 5 \to 2 \to 6 \to 3 \to 4}$

5 → next = llrec (2 → 3 → 4, 6)   return in1 = 5 → 2 → 6 → 3 → 4

2 → next = llrec (6, 3 → 4)   return in1 = 2 → 6 → 3 → 4

6 → next = llrec (3 → 4, Null)   return in1 = 6 → 3 → 4

return 3 → 4

(end of recursion)

Question b:

in1 = nullptr   in2 = 2

llrec (nullptr, [2])

⇓

if (in1 == nullptr) == True   => return in2;

∴ in2 = [2]; is returned by llrec (in1, in2) when in1 = nullptr
in2 = [2]