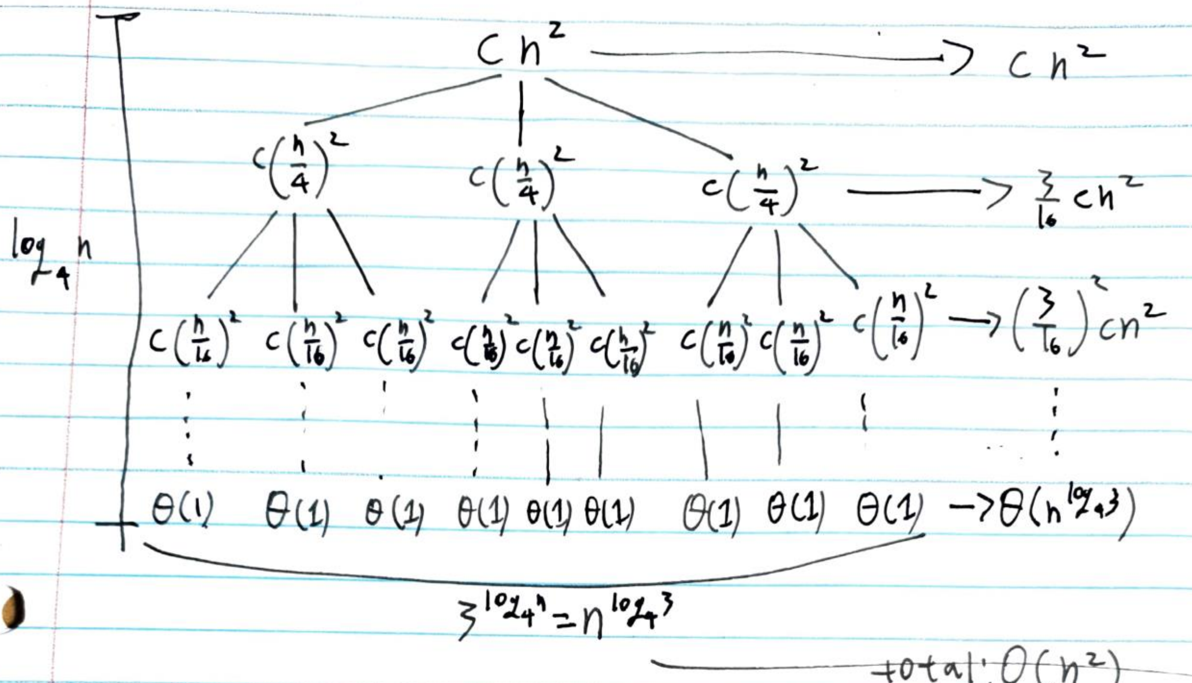
Q1 :

$T(n) = 3T(n/4) + \Theta(n^2)$



For simplicity, assume that n is power of 4 and the base case is $T(1) = \Theta(1)$, subproblem size for nodes at depth i is $n/4^i$. Get to base case when $n/4^i = 1 \Rightarrow n = 4^i \Rightarrow i = \log_4 n$.

Each level has 3 times as many nodes as the level above, so that depth i has 3^i nodes. Each internal node at depth i has cost $c(n/4^i)^2 \Rightarrow$ total cost at depth i (except for leaves) is $3^i c(n/4^i)^2 = (3/16)^i cn^2$. Bottom Level has depth $\log_4 n \Rightarrow$ number of leaves is $3^{\log_4 n} = n^{\log_4 3}$. Since each leaves contributes $\Theta(1)$, total cost of leaves $\Theta(n^{\log_4 3})$.

Add up costs over all levels to determine cost for the entire tree:

$$T(n) = \sum_{i=0}^{\log_4 n} \left(\frac{3}{16}\right)^i cn^2 + \Theta(n^{\log_4 3}) < \sum_{i=0}^{\infty} \left(\frac{3}{16}\right)^i cn^2 + \Theta(n^{\log_4 3})$$

$$= \underbrace{\frac{1}{1-(3/16)}}_{\text{decreasing geometric series}} cn^2 + \Theta(n^{\log_4 3}) = \frac{16}{13}cn^2 + \Theta(n^{\log_4 3}) = O(n^2)$$

substitute method

Use substitution method to verify $O(n^2)$ upperbound.

Show that $T(n) \leq d n^2$ for constant $d > 0$.

$$T(n) \leq 3T(n/4) + cn^2$$
$$\leq 3T(n/4)^2 + cn^2$$
$$= \frac{3}{16} d n^2 + cn^2$$
$$\leq d n^2 \quad \text{by choosing } d \geq (16/13)c.$$

That gives an upperbound of $O(n^2)$. The lower bound of $\Omega(n^2)$ is obvious because the recurrence contains a $\theta(n^2)$ term.

Hence, $T(n) = \theta(n^2)$

Q2

(a). $\because$ The higher the ~~hight~~ height of tree $H(n)$, the longer the exection time $T(n)$

$\therefore$ $\max[H(n)] \equiv \max[T(n)]$, $\min[H(n)] \equiv \min[T(n)]$

$\because$ when $q = 0$ or $n-1$, we have $\max[H(n)] = n-1$

when $q = \frac{n}{2}$, we have $\min[H(n)] = \Theta \lg n$

$\therefore$ $\max[T(n)] = T(n-1) + \Theta(n)$

$\min[T(n)] = T(\frac{n}{2}) + T(\frac{n}{2}) + \Theta(n)$

For ~~wst~~ worst case, we assume $T(n) \leq cn^2$, we have

$\max[T(n)] = c(n-1)^2 + \Theta(n) = c(n^2-2n+1) + \Theta(n)$

$\max[T(n)] = cn^2 - (2n-1) + \Theta(n) \leq cn^2 = O(n^2)$ when $c(2n-1) \geq \Theta(n)$

$\geq cn^2 = \Omega(n^2)$ when $c(2n-1) \leq \Theta(n)$

For best case, we assume $T(n) \leq cn\lg n$. we have

$\min[T(n)] = c\frac{n}{2}(\lg n - 1) + c\frac{n}{2}(\lg n - 1) + \Theta(n)$

$\min[T(n)] = cn\lg n - cn + \Theta(n)$, $\leq cn\lg n = O(n\lg n)$ when $cn \geq \Theta(n)$

$\geq cn\lg n = \Omega(n\lg n)$ w/m $cn \leq \Theta(n)$

B

RANDOMIZED-PARTITION(A, p, r)

$i \leftarrow$ RANDOM(p, r)

exchange $A[r] \leftrightarrow A[i]$

return PARTITION(A, p, r)

RANDOMIZED-QUICKSORT(A, p, r)

if $p < r$

  then $q \leftarrow$ RANDOMIZED-PARTITION(A, p, r)

    RANDOMIZED-QUICKSORT$(A, p, q-1)$

    RANDOMIZED-QUICKSORT$(A, q+1, r)$

Average-case analysis

- The dominant cost of the algorithm is partitioning.
- PARTITION removes the pivot element from future consideration each time.
- Thus, PARTITION is called at most n times.
- QUICKSORT recurses on the partitions.
- The amount of work that each call to PARTITION does is a constant plus the number of comparisons that are performed in its **for** loop.
- Let $X =$ the total number of comparisons performed in all calls to PARTITION.
- Therefore, the total work done over the entire execution is $O(n + X)$.

We will now compute a bound on the overall number of comparisons.

For ease of analysis:

- Rename the elements of A as $z_1, z_2, \ldots, z_n$, with z_i being the ith smallest element.
- Define the set $Z_{ij} = \{z_i, z_{i+1}, \ldots, z_j\}$ to be the set of elements between z_i and z_j, inclusive.

Each pair of elements is compared at most once, because elements are compared only to the pivot element, and then the pivot element is never in any later call to PARTITION.

Let $X_{ij} = \mathrm{I}\{z_i \text{ is compared to } z_j\}$.
(Considering whether z_i is compared to z_j at any time during the entire quicksort algorithm, not just during one call of PARTITION.)

Since each pair is compared at most once, the total number of comparisons performed by the algorithm is

$$X = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij} .$$

Take expectations of both sides, use Lemma 5.1 and linearity of expectation:

$$\begin{aligned}
\mathrm{E}[X] &= \mathrm{E}\left[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}\right] \\
&= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathrm{E}[X_{ij}] \\
&= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \Pr\{z_i \text{ is compared to } z_j\} .
\end{aligned}$$

- Once a pivot x is chosen such that $z_i < x < z_j$, then z_i and z_j will never be compared at any later time.
- If either z_i or z_j is chosen before any other element of Z_{ij}, then it will be compared to all the elements of Z_{ij}, except itself.
- The probability that z_i is compared to z_j is the probability that either z_i or z_j is the first element chosen.
- There are $j-i+1$ elements, and pivots are chosen randomly and independently. Thus, the probability that any particular one of them is the first one chosen is $1/(j-i+1)$.

Therefore,

$$\begin{aligned}
\Pr\{z_i \text{ is compared to } z_j\} &= \Pr\{z_i \text{ or } z_j \text{ is the first pivot chosen from } Z_{ij}\} \\
&= \Pr\{z_i \text{ is the first pivot chosen from } Z_{ij}\} \\
&\quad + \Pr\{z_j \text{ is the first pivot chosen from } Z_{ij}\} \\
&= \frac{1}{j-i+1} + \frac{1}{j-i+1} \\
&= \frac{2}{j-i+1} .
\end{aligned}$$

Substituting into the equation for $\mathrm{E}[X]$:

$$\mathrm{E}[X] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1} .$$

Evaluate by using a change in variables ($k = j - i$) and the bound on the harmonic series in equation (A.7):

$$\begin{aligned}
\mathrm{E}[X] &= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1} \\
&= \sum_{i=1}^{n-1} \sum_{k=1}^{n-i} \frac{2}{k+1} \\
&< \sum_{i=1}^{n-1} \sum_{k=1}^{n} \frac{2}{k} \\
&= \sum_{i=1}^{n-1} O(\lg n) \\
&= O(n \lg n) .
\end{aligned}$$

So the expected running time of quicksort, using RANDOMIZED-PARTITION, is $O(n \lg n)$.

Q3.

a). From 'Algorithm(A, B, n, k)', we have $T_a(n) = \Theta(n+k)$.

$\because$ Radix-sort(A, d) do d times 'Algorithm(A, B, n, k)'.

$\therefore$ Radix-sort $T_r(n) = \Theta(d(n+k))$.

$\because$ b = bits/word, r = bits/digit. $\therefore$ $d = ceil(b/r)$

$\because$ ~~k is the max key value of~~ $k = 2^r - 1$

3

$\therefore$ $T_r(n) = \Theta(\frac{b}{r}(n+2^r-1)) = \Theta(\frac{b}{r}(n+2^r))$

b). in order to let $T(n) = \Theta(\frac{b}{r}(n+2^r)) = \Theta(bn/\lg n)$

we choose $r = \lg n$. then we have

$T(n) = \Theta(\frac{b}{\lg n}(n+n)) = \Theta(\frac{bn}{\lg n})$

Q4:

$N = 13$.

a) $h(x) = x \bmod 13$

$h(73) = 8$. $h(31) = 5$ $h(32) = 6$ $h(59) = 7$

$h(44) = 5$, $h(22) = 9$. $h(41) = 2$ $h(18) = 5$

0	1	2	3	4	5	6	7	8	9	10	11	12
		41			31	32	59	73	22			
					44							
					18							

b) $Prob_s(x)$ $= x \bmod 13 + d$ (# steps until reach empty space)

$P(73) = 8 + 0$ $P(31) = 5 + 0$

$P(32) = 6 + 0$. $P(59) = 7 + 0$ $P(44) = 5 + 4 = 9$

$P(22) = 9 + 1 = 10$ $P(41) = 2 + 0$ $P(18) = 5 + 6 = 11$

0	1	2	3	4	5	6	7	8	9	10	11	12
		41			31	32	59	73	44	22	18	

c). $h(x) = x \bmod 13$ $\qquad q = 11$

$d(x) = 11 - x \bmod 11$

$P(x) = (h(x) + i * d(x))$

i: # of iteration

x	$h(x)$	$d(x)$	Probs
73	8	4	8
31	5	2	5
32	6	1	6
59	7	7	7
44	5	11	16 → (16 − 13) = 3
22	9	11	9
41	2	3	2
18	5	4	0 → (5 + 4) = 9 → (9 + 4) = 13 → 0

16

0	1	2	3	4	5	6	7	8	9	10	11	12
18		41	44		31	32	59	73	22			

Q5:

Define a random variable:

- n_i = the number of elements placed in bucket $B[i]$.

Because insertion sort runs in quadratic time, bucket sort time is

$$T(n) = \Theta(n) + \sum_{i=0}^{n-1} O(n_i^2) .$$

Take expectations of both sides:

$$\begin{aligned}
\mathrm{E}[T(n)] &= \mathrm{E}\left[\Theta(n) + \sum_{i=0}^{n-1} O(n_i^2)\right] \\
&= \Theta(n) + \sum_{i=0}^{n-1} \mathrm{E}[O(n_i^2)] && \text{(linearity of expectation)} \\
&= \Theta(n) + \sum_{i=0}^{n-1} O(\mathrm{E}[n_i^2]) && (\mathrm{E}[aX] = a\mathrm{E}[X])
\end{aligned}$$

Claim

$\mathrm{E}[n_i^2] = 2 - (1/n)$ for $i = 0, \ldots, n-1$.

Proof of claim

Define indicator random variables:

- $X_{ij} = \mathrm{I}\{A[j] \text{ falls in bucket } i\}$
- $\Pr\{A[j] \text{ falls in bucket } i\} = 1/n$
- $n_i = \sum_{j=1}^{n} X_{ij}$

Then

$$\begin{aligned}
\mathrm{E}[n_i^2] &= \mathrm{E}\left[\left(\sum_{j=1}^{n} X_{ij}\right)^2\right] \\
&= \mathrm{E}\left[\sum_{j=1}^{n} X_{ij}^2 + 2\sum_{j=1}^{n-1}\sum_{k=j+1}^{n} X_{ij}X_{ik}\right] \\
&= \sum_{j=1}^{n} \mathrm{E}[X_{ij}^2] + 2\sum_{j=1}^{n-1}\sum_{k=j+1}^{n} \mathrm{E}[X_{ij}X_{ik}] && \text{(linearity of expectation)}
\end{aligned}$$

$$
\begin{aligned}
\mathrm{E}\left[X_{ij}^2\right] &= 0^2 \cdot \Pr\{A[j] \text{ doesn't fall in bucket } i\} + 1^2 \cdot \Pr\{A[j] \text{ falls in bucket } i\} \\
&= 0 \cdot \left(1 - \frac{1}{n}\right) + 1 \cdot \frac{1}{n} \\
&= \frac{1}{n}
\end{aligned}
$$

$\mathrm{E}[X_{ij} X_{ik}]$ for $j \neq k$: Since $j \neq k$, X_{ij} and X_{ik} are independent random variables

$$
\begin{aligned}
\Rightarrow \mathrm{E}[X_{ij} X_{ik}] &= \mathrm{E}[X_{ij}]\,\mathrm{E}[X_{ik}] \\
&= \frac{1}{n} \cdot \frac{1}{n} \\
&= \frac{1}{n^2}
\end{aligned}
$$

Therefore:

$$
\begin{aligned}
\mathrm{E}[n_i^2] &= \sum_{j=1}^{n} \frac{1}{n} + 2 \sum_{j=1}^{n-1} \sum_{k=j+1}^{n} \frac{1}{n^2} \\
&= n \cdot \frac{1}{n} + 2 \binom{n}{2} \frac{1}{n^2} \\
&= 1 + 2 \cdot \frac{n(n-1)}{2} \cdot \frac{1}{n^2} \\
&= 1 + \frac{n-1}{n} \\
&= 1 + 1 - \frac{1}{n} \\
&= 2 - \frac{1}{n} \qquad \blacksquare \text{ (claim)}
\end{aligned}
$$

Therefore:

$$
\begin{aligned}
\mathrm{E}[T(n)] &= \Theta(n) + \sum_{i=0}^{n-1} O(2 - 1/n) \\
&= \Theta(n) + O(n) \\
&= \Theta(n)
\end{aligned}
$$