1. - D_i be the heap after the i^{th} operation, and let D_i consist of n_i elements

- At most time taken by INSERT or EXTRACT-MIN operation — $k \log n$ time. Here k is constant and $n = \max(n_{i-1}, n_i)$.

$\therefore$
$$D_i = \begin{cases} 0 & \text{if } n_i = 0 \\ k(n_i \log n_i) & \text{if } n_i > 0 \end{cases}$$

Let $d_i(x)$ be the depth of node x in D_i

$$\phi(D_i) = \sum_{x \in D_i} k(d_i(x) + 1) = k\left(n_i + \sum_{x \in D_i} d_i(x)\right)$$

- When heap is empty $\rightarrow$ $\phi(D_i) = 0$ and we always have $\phi(D_i) >= 0$

* INSERT OPERATION :

- After an INSERT operation, the sum changes only by an amount equal to the depth of the new last node $d_i(n)$ of the heap, which is $[\log n_i]$
- The change in potential due to an INSERT is $k(1 + [\log n_i])$
- Therefore, the amortized cost of INSERT is;

$$O(\log n_i) + O(\log n_i) = O(\log n_i)$$

$$= O(\log n)$$

* EXTRACT-MIN Operation :

- After EXTRACT-MIN, the sum changes by the negative of the depth of the old node in heap
- So, there is a decrease in the potential by $k(1 + [\log n_i])$
- Thus, the amortized cost is at most ;

$$(k \log n_i) - k(1 + [\log n_i]) = \underline{\underline{O(1)}}$$

2] $$\phi(T) = \sum_{i=1}^{k} |size(T_i) - num(T_i) - N| \quad \text{—— ①}$$

Answer :-

ⓐ Prove that the contribution of table T_i to $\phi(T)$ is zero immediately after it expands or contracts

→ Given : $size(T_i') = num(T_i') + N'$ —— ②

∴ From ①,

$$(T_i') = |size(T_i') - num(T_i') - N'|$$

Substituting value for $size(T_i')$ from line ②

$$(T_i') = |num(T_i') + N_i' - num(T_i') - N'|$$

$$\therefore (T_i') = 0$$

Hence proved.

(b) Prove that the amortized cost of an insertion that does not require an expansion is at most $k + O(1)$

→ Let T_j be the new table

Actual cost 'c' without any insertion $- O(1)$ —— ①

$$\phi(T') - \phi(T) =$$

$$\sum_{\substack{i=1 \text{ and } i \neq j \\ (A)}}^{k} ((num(T_i') + N' - size(T_i')) -$$

$$(num(T_i') + N' - size(T_i'))) +$$

$$((num(T_j') + N' - size(T_j')) -$$

$$(num(T_j') + N' - size(T_j')))$$

$$= (k-1) + 2 = k \quad \text{—— ②}$$

Therefore, the amortized cost of an insertion that does not require an expansion is;

$$\hat{c} = c + \phi(T') - \phi(T)$$

$$= O(1) + k \quad \text{—— from ① \& ②}$$

(c) Let T_j be the new table

The amortized cost of an insertion to table T_j requiring an expansion is ;

$$\hat{c} = c + \phi(T') - \phi(T)$$

$$= O(N) + \sum_{i=1 \wedge i \neq j}^{k} ((num(T_i') + N' - size(T_i')) - (num(T_i) + N - size(T_i))) + ((num(T_j') + N' - size(T_j')) - (num(T_j) + N - size(T_j)))$$

$$= O(N) + (k-1) - N$$

$$= \underline{\underline{O(k)}}$$

Hence, proved.

Date ______ /

(d) (i) No contraction

$$\phi(T') = \sum_{i=1 \wedge i \neq j}^{k} (size(T'_i) - num(T'_i) - N') + (size(T'_j) - num(T'_j) - N') \quad \text{--- (1)}$$

$$\phi(T) = \sum_{i=1 \wedge i \neq j}^{k} (size(T_i) - num(T_i) - N) + (size(T_j) - num(T_j) - N) \quad \text{--- (2)}$$

$\therefore$ Amortized cost

$$\hat{c} = c + \phi(T') - \phi(T)$$

$$= O(1) + (k-1) + 2 \quad \text{--- from (1) \& (2)}$$

$$= \underline{\underline{O(k)}}$$

(ii) The table in which deletion occurs is also contracted

- The amortized cost of a deletion is $O(k)$
- That means total 'k' number of tables are contracted
- So, let S be the set of indexes of tables that contract & $s = |S|$

$\therefore$ Amortized cost $\hat{c} = c + \phi(T') - \phi(T)$

$$= O(N) + \sum_{i \notin S} ((size(T_i') - num(T_i') - N') - (size(T_i) - num(T_i) - N)) + \sum_{i \in S} ((size(T_i') -$$

Page No. ______
Date ______

$num(T_i') - N') - (size(T_i) - num(T_i) - N))$

$= O(N) + (k-s) - \sum_{i \in S} (size(T_i) - num(T_i) - N)$

Given:

Any table T_i that has been contracted satisfies $size(T_i') = num(T_i') + N$. Substituted this equation in the above amortized equation

- If $num(T_i) \leq size(T_i) - 2N + 2$, the element is deleted ~~then the element is simply~~ and then T_i is contracted to size $num(T_i) + N - 2$ by copying the elements in all the tables into a smaller region.

- As $i \in S$, $num(T_i) \leq size(T_i) - 2N + 2$

  i.e. $size(T_i) - num(T_i) - N \geq N - 2$

$\therefore \hat{c} \leq O(N) + (k-s) - s(N-2)$

$= \underline{\underline{O(k)}}$

(iii) Some tables contract but the table that is the site of deletion is not one of them

→ - Actual cost 'c' $= O(N)$ ——— ①

$\hat{c} = c + \phi(T') - \phi(T)$

Page No. ______
Date ______

$$\phi(T') = \sum_{i \notin S} (size(T_i') - num(T_i') - N') +$$

$$\sum_{i \in S/\{j\}} (size(T_i') - num(T_i') - N') +$$

$$(size(T_j') - num(T_j') - N')$$

$$= (k-s) + 2 \qquad \text{——— (2)}$$

$$\phi(T) = \sum_{i \notin S} (size(T_i) - num(T_i) - N) +$$

$$\sum_{i \in S/\{j\}} (size(T_i) - num(T_i) - N) +$$

$$(size(T_j) - num(T_j) - N)$$

$$= (s-1)(N-2) \qquad \text{——— (3)}$$

$\therefore$ From 1, 2 & 3 ; Amortized cost will be

$$\hat{c} = c + \phi(T') - \phi(T)$$

$$= O(N) + (k-s) + 2 - (s-1)(N-2)$$

$$= \underline{\underline{O(k)}}$$

Page No. ______
Date ______

2. (e) Consider all the tables are empty and no space is allocated ;

$\therefore$ $size(T_i) = num(T_i) = 0$

$\{ \text{for } 1 \leq i \leq k \}$; $\forall T_i \in T$

- Insert k elements into table sequencially
- Time taken to relocate tables for i^{th} insertion will take $i-1$ time
- Therefore, the sequence of insertion takes ;

$$\sum_{i=1}^{k} (1 + i - 1) + k = \underline{\underline{k(k+1)/2}}$$

- Deleting all elements from table will take time k as $size(T_i) = num(T_i) = 0$ after the sequence of deletions.

$\therefore$ Sequence of insertions and deletions

$$k(k+1)/2 + k$$

- Repeating the above for n times gives

$$= n(k(k+1)/2 + k)$$

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

$$= nk\left(\frac{k+1}{2} + 1\right)$$

$$= nk\left(\frac{k+1+2}{2}\right)$$

$$= nk\left(\frac{k+3}{2}\right)$$

= Let nk be m i.e., $m = nk$, then

$$= m\,((k+3)/2)$$

$$= O(mk) \cong \Omega(km)$$

Date ____________

(f) Given :

$$size(T_i) \leq num(T_i) + 2N$$

TO prove all tables always satisfy above condition, consider following proofs ;

(i) When all tables are empty and no space is allocated, then for each table (T_i)

$$size(T_i) = num(T_i) + 2N = 0$$

(ii) When table T_i is expanded / contracted :

$$size(T_i') = num(T_i') + N' \leq num(T_i') + 2N'$$

(iii) When an insertion occurs in T_i but T_i doesn't expand then

$$size(T') = size(T_i) \leq num(T_i) + 2N < num(T_i') + 2N'$$