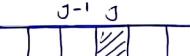


4. while $k < A[i]$ and $i > \sum_{j=2}^n t_j$ do :	$C_4$
5. $A[i+1] \leftarrow A[i]$	$\sum_{j=2}^n C_5 + t_j$
6. $i \leftarrow i-1$	$\sum_{j=2}^n C_6 + t_j$
7. $A[i+1] \leftarrow \text{key}$	$C_7 n-1$
$t_j = \text{while}$ $\text{تمام خود} \rightarrow \text{تمام خود}$	$\text{تمام خود} \rightarrow \text{تمام خود}$
for $\text{تمام خود}$ $\rightarrow \text{تمام خود}$	$\text{تمام خود} \rightarrow \text{تمام خود}$
$T(n) = C_1 n + C_2(n-1) + C_3(n-1) + C_4(\sum_{j=2}^n t_j + 1)$	$\text{تمام خود} \rightarrow \text{تمام خود}$
$+ C_5 \sum_{j=2}^n t_j + C_6 \sum_{j=2}^n t_j + C_7(n-1)$	$\text{تمام خود} \rightarrow \text{تمام خود}$
$\therefore t_j \leq j-1$	
$\min(T(n)) = \alpha \cdot n + b$	
$(t_j = 0)$	
$\max(T(n)) =$	
$(t_j = j-1)$	
$\sum_{j=2}^n t_j = \sum_{j=2}^n j-1 = \sum_{j=1}^{n-1} j = \frac{(n-1)(1+n-1)}{2}$	
$= \frac{(n-1)(1+n-1)}{2} = \frac{n(n-1)}{2}$	

ناهال  
NAHAL

Subject: _____	Date: _____
$\therefore (\text{Insertion sort})$	$\text{تحليل الソート مرتب جدول دفعات}$
$A$ 	$\text{أ عدد صفوف درجات} \rightarrow \text{input}$
	$\text{A}[i:j] \rightarrow \text{output}$
	$\text{هي صورت مرتب شده مثلاً}$
	$A[i:j] \rightarrow A[i+1:n]$
	$\forall i \in [n]$
	$\therefore (\text{Incremental})$ $\rightarrow$ $\text{روتين احترازي}$
	
	$\text{خانة } j \text{ اما رابط } (i-j) \text{ امام}$
	$\text{هذا يسمى دفعات مرتب}$
	$\Rightarrow \text{خانة } i \text{ بعده يسمى دفعات مرتب}$
	$\therefore \text{تحليل الソート مرتب جدول دفعات}$
	$\text{Insertion sort } (A, n) :$
1. $\text{for } j=2 \text{ to } n \text{ do:}$	$C_1 n$
2. $\text{key} \leftarrow A[j]$	$C_2 n-1$
3. $i \leftarrow j-1$	$C_3 n-1$

ناهال  
NAHAL

Subject: \_\_\_\_\_  
Year: \_\_\_\_\_ Month: \_\_\_\_\_

Bubble-Sort(A, n) :

1. for  $i \leftarrow (n-1)$  down to 1 do:
  - $\sum_{i=1}^{n-1} c_1$
2. for  $j \leftarrow 0$  to  $i$  do:
  - $\sum_{i=1}^{n-1} c_2$
3. if  $A[j] > A[j+1]$  then:
  - swap( $A[j], A[j+1]$ )
4.  $\sum_{i=1}^{n-1} c_3$

مازنیدم هزینه ۴، ۳ است یعنی غریب نیست سرت سرت برقوار است.

$$T(n) \leq n c_1 + c_2 \sum_{i=1}^{n-1} (i+1) + c_3 \sum_{i=1}^{n-1} i$$

$$= c_1 \cdot n + c_2 \frac{(n+2)(n-1)}{2} + c_3 \frac{n(n-1)}{2} = an^2 + bn + c$$

$$\Rightarrow a' n^2 + b' n + c' \rightarrow \text{مازنیدم تابع ۲ نیست.}$$

$$\max(T(n)) = c_1 n + c_2 (n-1) + c_3 (n-1) + c_4 (n+2) \\ (n-1)/2$$

$$= c_5 n(n-2)/2 + c_6 n(n-1)/2 + c_7 (n-1)$$

$$= a' n^2 + b' n + c$$

درین حالت رایج درجه ۲ مربوط شده کی تقریر است.

$$\therefore T(n) = a' n^2 + b' n + c'$$

(Bubble Sort) :  $y_{40} y_{50}$  مرتب

Bubble-Sort(A, n) :

1. for  $i \leftarrow 2$  down to 1 do:
  - for  $j \leftarrow 0$  to  $i$  do:
    - if  $A[j] > A[j+1]$  then:
      - swap( $A[j], A[j+1]$ )

دش دهان باز پرداز که نیزه رین  
مک زنی بر لین بز ب لین

Subject : \_\_\_\_\_  
Year. Month.

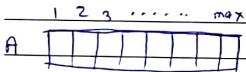
Queue داده صف ساخته شد

First In , First Out  $\leftarrow$  FIFO

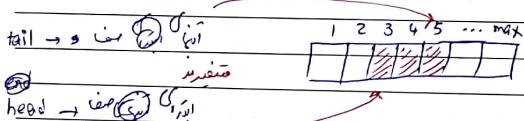
Method ۱ - Add(Q, x)  $\rightarrow$  اضافه کردن  
- Del(Q)  $\rightarrow$  حذف کردن  
Q اضافه

- Find(Q, x)  $\rightarrow$  ۱) x in Q or not

- Is Empty(Q)  $\rightarrow$ , - Is Full(Q)



سیده ملک  $\rightarrow$  Add(Q, x)



head  $\rightarrow$  ابتداء

SINA



Subject : \_\_\_\_\_  
Year. Month.

$f \in O(g)$  ①  $a < b$  ①

$f \in \Omega(g)$  ②  $a > b$  ②

$f \in \Theta(g)$  ③  $a = b$  ③

$\leftarrow$  little O :  $O(n)$

$f(n) \in O(g(n)) \Leftrightarrow \forall c > 0 \exists n_0 : f(n) < c g(n)$

$$f(n) \begin{cases} 2n+5 \in O(n^2) \\ 2n+5 \in O(\log(n)) \\ 2n+5 \in O(n) \end{cases} \quad \lim_{n \rightarrow \infty} \frac{g(n)}{f(n)} = \infty$$

$\leftarrow$  little Omega :  $\Omega(n)$

$f(n) \in \omega(g(n)) \Leftrightarrow \forall c > 0 \exists n_0 : f(n) > c g(n)$

$f(n) > c g(n)$

مسیر  $g(n)$  تحریز است.

$$\lim_{n \rightarrow \infty} \frac{g(n)}{f(n)} = \infty \quad \Leftrightarrow \begin{cases} 2n+5 \in \omega(\sqrt{n}) \\ 2n+5 \in \omega(\log n) \\ 2n+5 \notin \omega(n) \\ 2n+5 \notin \omega(n^2) \end{cases}$$

SINA

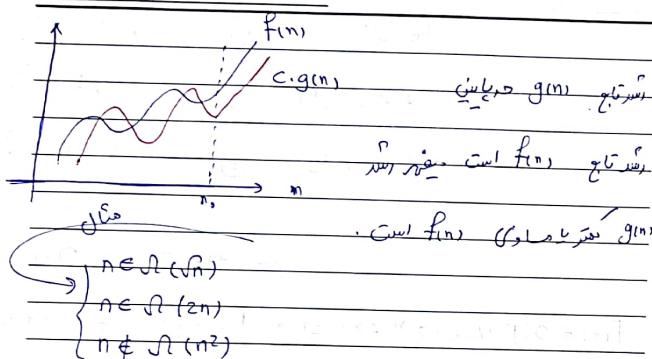
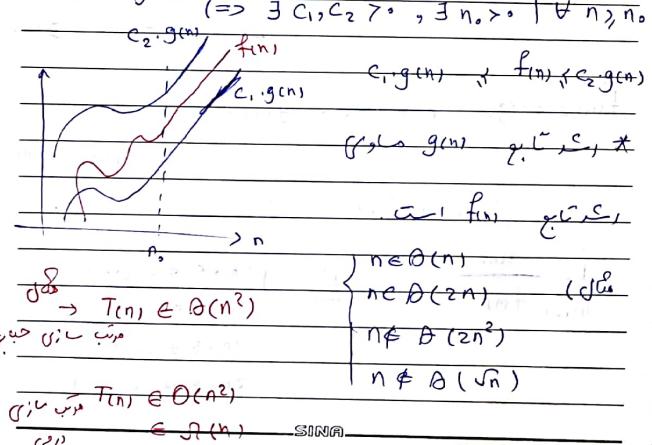
ISE

1.  $f(n) \in \Omega(g(n)) \Leftrightarrow \exists c > 0, \exists n_0 > 0 \forall n \geq n_0, f(n) \geq c \cdot g(n)$  $\leftarrow$  big omega:  $\Omega(n)$ 

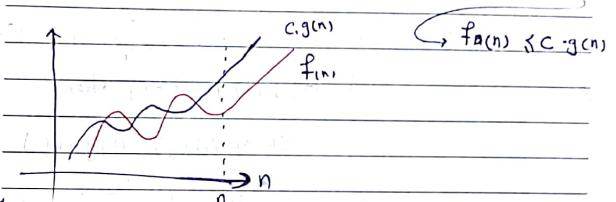
2.

Subject: \_\_\_\_\_  
Year: \_\_\_\_\_ Month: \_\_\_\_\_

3.

 $f(n) \in \Theta(g(n)) \Leftrightarrow \exists c_1, c_2 > 0, \exists n_0 > 0 \forall n \geq n_0, c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$  $T(n) \in O(n^2) \in \Omega(n)$ Subject: \_\_\_\_\_  
Year: \_\_\_\_\_ Month: \_\_\_\_\_3.  $\mathcal{E}^{O(n)}$ 

1/12

 $\therefore$  big O :  $O(n)$  $f(n) \in O(g(n)) \Leftrightarrow \exists c > 0, \exists n_0 > 0 \forall n \geq n_0,$ 

برهان عوقب برقرار رسانید  $\sum_{i=1}^n i^2$   $\leq \frac{n}{2}(n+1)^2$

$\sum_{i=1}^n i^2 \leq \frac{n}{2}(n+1)^2$   $\leq \frac{n}{2}(n^2 + 2n + 1) = \frac{n}{2}n^2 + n^2 + \frac{n}{2}$

$\therefore \sum_{i=1}^n i^2 \leq \frac{3}{2}n^3$   $\leftarrow O(n^3)$

$n \in O(n^2)$   $\therefore$   $n^2$  است.  $g(n)$

$n \in O(2n)$

$n \notin O(\sqrt{n})$

 $n \in O(2n) \Leftrightarrow \exists c > 0, \exists n_0 > 0 \forall n \geq n_0, C \cdot 2n$ 

$n \leq C \cdot 2n \Rightarrow C \geq \frac{1}{2}$

$C \geq \frac{1}{2} \Rightarrow n_0 = 1$

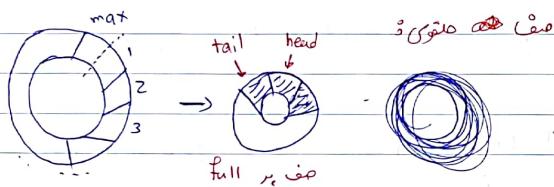
$n \in O(n^2) \rightarrow n \leq C \cdot 2n^2 \rightarrow 2n^2 - n \geq 0$

$\therefore n_0 = 1$

IsEmpty ( $Q$ ):

1. if  $head > tail$  then:  
    return true
- 2.
3. else return false

Initialization:  $\{ head = 1 \rightarrow \text{نقطة ابتداء} \quad head = 1$   
 $tail = 0 \rightarrow \text{نقطة انتهاء}$



نقطة ابتداء

IsFull ( $Q$ ):

1. if  $head == tail == max$ :  
    return true
- 2.
3. else return false

head = 1  $\rightarrow$  *نقطة ابتداء*

tail = 0  $\rightarrow$  *نقطة انتهاء*

head = 1  $\rightarrow$  *نقطة ابتداء*

tail = 1  $\rightarrow$  *نقطة انتهاء*

Empty  $\rightarrow head = tail + 1$

Full  $\rightarrow head = tail + 1$

Add:  $Q[tail] = x \rightarrow tail++$

Del:  $x = Q[head] \rightarrow head++$

نقطة ابتداء

Add ( $Q, x$ ):

- Subject: \_\_\_\_\_  
Year: \_\_\_\_\_ Month: \_\_\_\_\_ Date: \_\_\_\_\_
1. if  $tail == max$  then:  
    error ("Queue is full")
  - 2.
  3. return
  4. else:  
     $tail = tail + 1$
  - 5.
  6.  $Q[tail] \leftarrow x$

Del ( $Q$ ):

1. if  $head > tail$  then:  
    error ("Queue is empty")
- 2.
3. return
4. else:  
     $x = Q[head]$
5.  $head = head + 1$
6. return  $x$

Find ( $Q, x$ ):

1. for  $i \leftarrow head$  to  $tail$  do:  
    if  $Q[i] == x$  then  
        return true
- 2.
- 3.
4. return False

نحوه از عملگر دو رک از عملگر top نمایه داریم و در کارهای بالا می‌بینیم

دارد آن نمایه عملگر بالا می‌بینیم pop نمایه داریم خوبی نمایه می‌بینیم

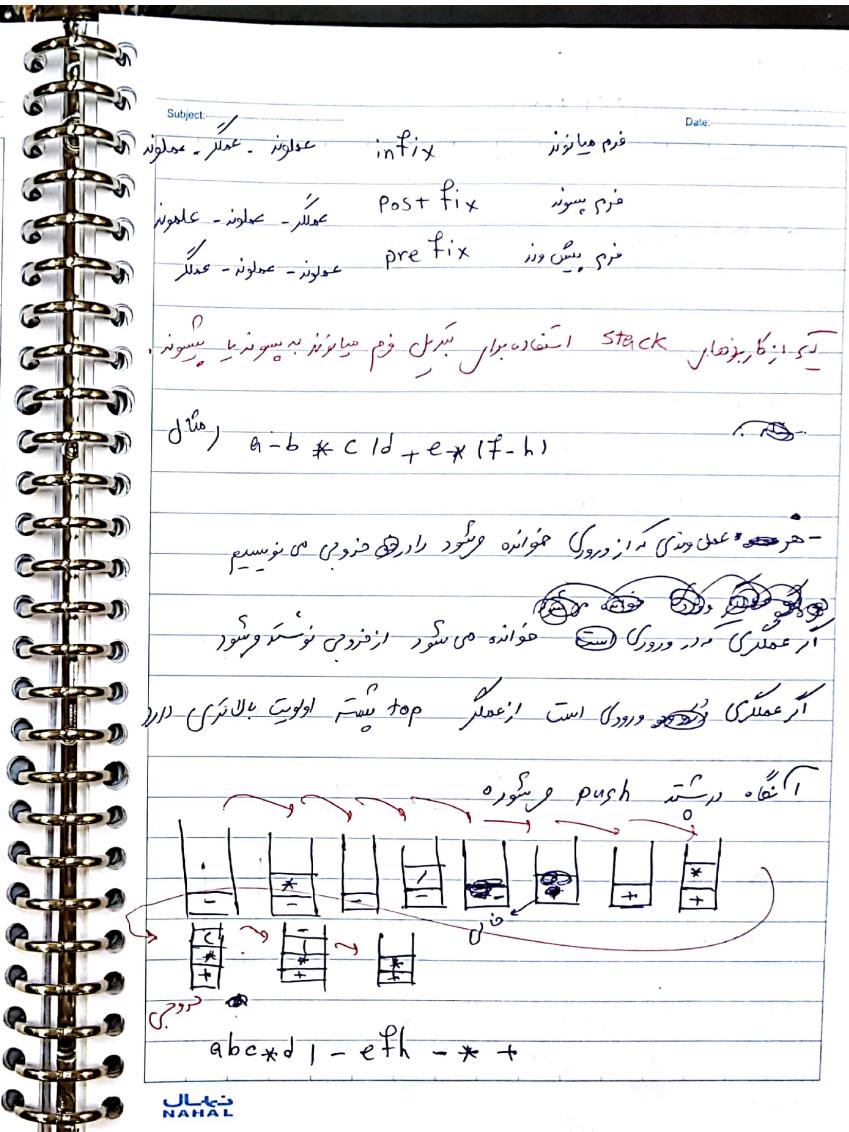
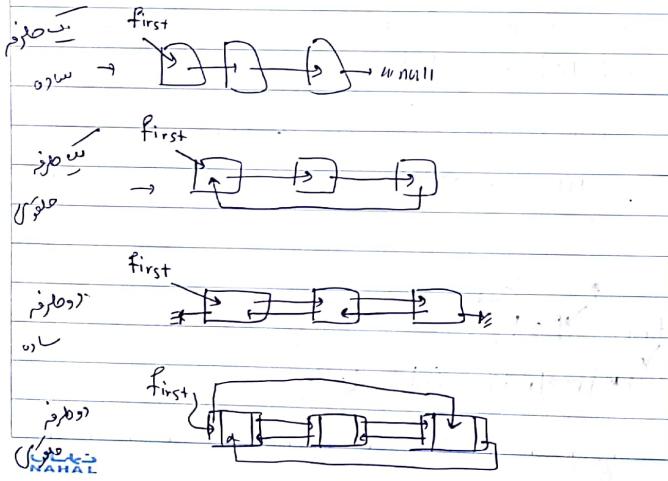
- پرانتز باز - ) - همین درست - push -

آخر عملگر دوری ) - ( - پرانتز بسته به فاعل عملگرها می‌بینیم تابع ) )

برایم pop نمایه داریم خوبی نمایه می‌بینیم خود - ) - راهم

همم کرده و متوجه نداشیم.

### لیست لیست‌ها (linked lists)



$\text{pop}(S)$ :

if  $S \neq \emptyset$  then:

error ("Stack is empty")

return

$x = S[\text{top}]$

$\text{top} = \text{top} - 1$

Find ( $S, x$ ) :

for  $i = 1$  to  $\text{top}$  do

if  $S[i] == x$  then:

return true

return false

: stack ✓ (برابر)

① فراخوانی توابع :

function A() {

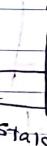
    function B() {

        call B()

        return

}

    return



↑ (برابر) stack  
مانند (برابر) A  
↑ (برابر) stack  
مانند (برابر) B

نیال NAHAL

LIFO

← Stack سینه سیفی

Last in - First out



push → دادن س ایتم فریم Top Stack عصر باری

pop → Stack فریم IsEmpty دادن ایتم IsFull

IsEmpty( $S$ ):

1. if  $\text{top} == 0$  then:
2. return true
3. else return false

IsFull( $S$ ):

1. if  $\text{top} == \text{max}$  then:
2. return true
3. else return false

Push( $S, x$ ):

1. if IsFull( $S$ ) then:
2. error("stack is full")
3. return
4.  $\text{top} = \text{top} + 1$
5.  $S[\text{top}] = x$

نیال NAHAL

Subject \_\_\_\_\_ Date: \_\_\_\_\_

مقدار کوچکترین عمق تا دور میان از ریشه در  $h(T)$  نامیده می‌شود.

Tree       $\leftarrow$  Binary Search      (BST)

وَيُرِسُ رَوْتَ جَسَّيْفُورْ دَوْدَوْ : اِمْ لَهْسِ رَوْتَ بَاسْدَا وَلْ دَرْزَرْزَ

۲۰۱۷-۱۳۹۶-۰۸-۰۵

$T(n)$

Inorder-Tree-Walk( $n$ ).

7. if  $n = \text{null}$ :

## Inorder - Tree Walks (2 left)

3. print n.key

NAHAL

اسکرپت نور جیو  
 Subject: \_\_\_\_\_ / Nod & p = getNodel();  
 Add(first, p); p.info = 5; Date: \_\_\_\_\_

7. if  $\text{first} == \text{NULL}$  then:
    2.  $\text{first} = p$       10.  $p.\text{prev} = q$
    3.  $p.\text{next} = p$       11.  $\text{first} = p$
    4.  $p.\text{prev} = p$
  5. else:
    6.  $p.\text{next} = \text{first}$
    7.  $q = \text{first}.\text{prev}$
    8.  $\text{first}.\text{prev} = p$

مکروہی ملکے سرکار:

The diagram shows a node structure for a binary search tree. The node is represented as a rectangle containing the word "key". An arrow labeled "parent" points to the top of the node. Two arrows point away from the node: one to the left labeled "left child" and one to the right labeled "right child".

exp →

The diagram shows a binary search tree with 4 nodes. The root node contains the value "4". It has two children: a left child containing "2" and a right child containing "3". The node "2" has two children: a left child containing "1" and a right child containing "3". The node "3" has two children: a left child containing "2" and a right child containing "4". Arrows indicate the parent-child relationships.

$r(G^{\text{pr}}) \rightarrow \text{indegree}(V) = \phi$



Node \* q = first;

while (q) {

if (q -> info == 1) break;

q = q -> next;

}

if (q) {

p -> next = q -> next;

q -> next = p;

}

Node \* p = first;

حذف از سیستم:

if (first) {

first = first -> next;

free(p);

}

Struct Node {

Subject: \_\_\_\_\_ Date: \_\_\_\_\_

int info;

Node \* next;

Node \* getNode() {

Node \* p;

p = (Node \*) malloc(sizeof(struct Node));

return p;

}

Node \* ptr = first

سیاست سیستم:

while (ptr) {

printf("info=%d", ptr -> info);

ptr = ptr -> next;

}



حذف از سیستم:

برونو:

} p -> next = first;  
first = p;

غیر معرفی

Subject: Tree-Min(x):

1. while  $x.left \neq \text{Null}$
2.  $x = x.left$
3. return  $x$

لبریزی که  $x.right \neq x.left$  درجی قدرتی کوچکتر است  $\max$  (C++)

Tree-successor(x):

1. if  $x.right \neq \text{Null}$  then:
    2. return Tree-Min( $x.right$ )
  3. ~~copy~~  $y = x.p$
  4. while  $y \neq \text{Null}$  and  $x == y.right$ 
    5.  $x = y$
    6.  $y = y.p$
  7. return  $y$
- دست نظریه داشت  
درجهت نظریه داشت  
دست نظریه داشت

بتیمال  
NAME:

4. Inorder-Tree-Walk ( $x.right$ )

Subject: \_\_\_\_\_

Date: \_\_\_\_\_

//  
g.v.      root  
Search( $x, k$ ):

1. if  $x == \text{Null}$  or  $k == x.key$ :
2. return  $x$
3. if  $k < x.key$ :
4. return Search( $x.left, k$ )
5. return Search( $x.right, k$ )

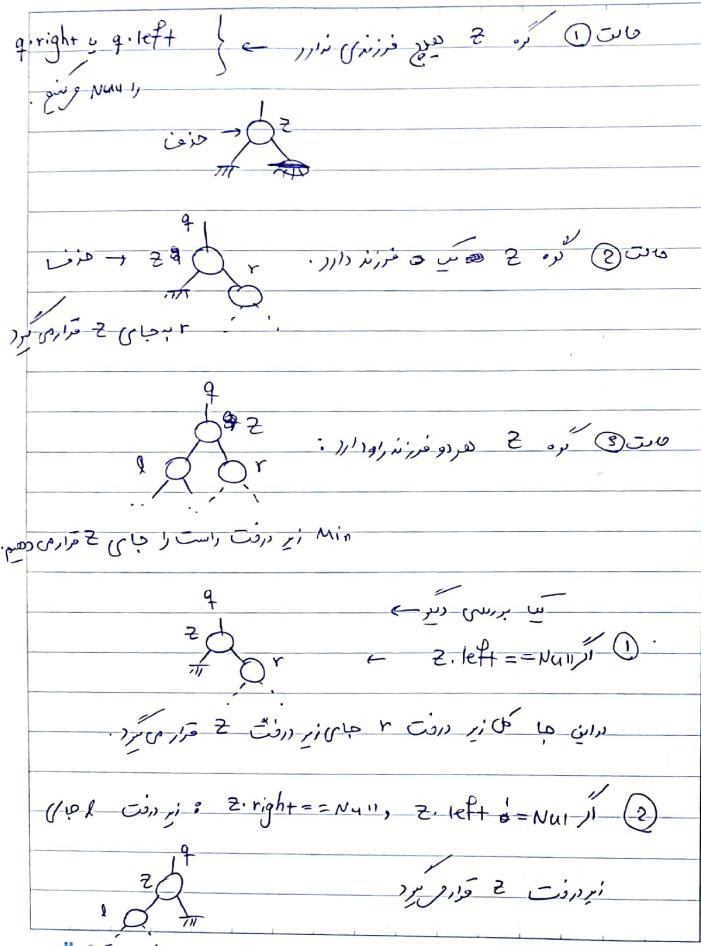
Iterative-search( $x, k$ ):

1. while  $x \neq \text{Null}$  and  $k \neq x.key$  do:
  2. if  $k < x.key$  then:
    3.  $x = x.left$
  4. else  $x = x.right$
5. return  $x$

Tree-Min( $x$ ):

1. if  $x == \text{NULL}$  or  $x.left == \text{NULL}$  then:
  2. return  $x$
3. return Tree-Min( $x.left$ )

Dekte → حذف



Binary Search Tree :

Subject \_\_\_\_\_ Date \_\_\_\_\_

حذف دفع

$z \leftarrow z \cdot key = value$

$z \cdot left = null$

$z \cdot right = null$

Insert / Delete

Tree-Insert ( $T, z$ ) :

1.  $y = null$

2.  $x = T.root$

3. while  $x \neq null$  do:

4.      $y = x$

5.     if  $z.key < x.key$  then

6.          $x = x.left$

7.     else  $x = x.right$

8.  $z.p = y$

9. if  $z.key < y.key$  then

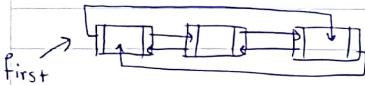
10.      $y.left = z$

11. else  $y.right = z$

مسن سوزن) (وطرفره مادوری)

Subject

first = سیخان



iterate:

print(first);

1. if first == NULL then:

2. return

3. p = first

4. do

5. print p.info

6. p = p.next

7. while p != first

Find(first, x):

1. if first == NULL:

2. return false

3. p = first

4. do

5. if p.info == x then,

7. p = p.next

8. while p != first

9. return false

6. return true;

ناهال NAHAL

DS-2018

Subject

حذف از جای خامن از بین

Date

Node \* p;

if (first) {

if (first->info == 11) {

p = first;

first = first->next;

free(p); }

else {

p = first;

while (p->next) {

if (p->next->info == 11) break;

p = p->next;

}

if (p->next) {

q = p->next;

p->next = p->next->next;

free(q);

}

ناهال NAHAL



لین ادعا با استوا کاپت من سود.

Subject: \_\_\_\_\_ Date: \_\_\_\_\_

$bh(x) \geq bh(y) + 1$

اربعن سده فریز (۴) حداقل بیشتر از  
اربعن سده پر (۴) همچوین است.

مقدیق فریز استوا

$$\rightarrow n = n_1 + n_2 + 1 \geq (2^{bh(x)-1} - 1) + (2^{bh(x)-1}) + 1$$

$$= 2^{bh(x)} - 1 \quad (1)$$

$$= n \geq 2^{bh(x)} - 1 \geq 2^{bh(x)/2} - 1 \quad (2)$$

$$\Rightarrow h(x) \leq 2 \log(n+1)$$

Rotation

وئیس باین سرج برک را بهم معن زند.

ناهال

Subject: \_\_\_\_\_ Date: \_\_\_\_\_

سیدرن جستجوی (دودی) بدهد که این روت دارند: را فرزن سده دستور.

۱) هرگز خوب نیسته است. ۲) روت (ذن) سده است.

۳) ویرایش (۴) مرز باس (۵) هر برگ (۶) سده است. (۷) باره باره، هم مسیرها ساده از آن بروید.

ثابت برای تعداد هزار سده حساب (دارد).

۸) محل بی میر (زیر بی کاپ بگ) هر دوی ابر طبله میز دستورها

مرد باتابرد دیگر است.

۹) شماره خوب سده (۱۰) از روی کاپ بگ می شناسد.

خوب نیست.

۱۱) سیدرن Red-Black با هدف دادن درایم ارتعض حداقل ۲ $\log(n+1)$  است.

ادعا: هر دوی نزدیکی داشت بلطف دستور

ناهال

همچنین  $\rightarrow C \times \frac{n}{4} \times 1 + C \times \frac{n}{2} \times 2 + \dots + C \times 1 \times \log_2 n$   
 $= \sum_{i=1}^{\log_2 n} C \frac{n}{2^{i+1}} \times i = \frac{Cn}{2} \sum_{i=1}^{\log_2 n} \frac{i}{2^i} = O(n)$ 
  
 شدید

**Heap-Sort**

1.   
 result:  $[1|3|4|4|5|1] \rightarrow$  این آرایه را می سویم.  
 داده دخواه می خواهیم sort کرد

**Heap-Sort(A):**  
 1. Build-Max-Heap(A);  $\Rightarrow O(n)$   
 2. for  $i \leftarrow \lfloor \frac{n}{2} \rfloor$  down to 1 do:  $\Rightarrow T(i) = O(n \log n)$   
 3.  $c_3(n-1) \leftarrow \text{swap}(A[i], A[n])$   
 4.  $c_4(n-1) \leftarrow \text{max\_Heapify}(A[1:n-1], \text{HeapSize})$   
 5.  $c_5(n-1) \leftarrow \text{max\_Heapify}(A[1:n])$

6. if  $\text{largest} \neq i$ :  
~~Heapify( $A[i], \text{largest}$ )~~  $\rightarrow A[\text{largest}] \leftrightarrow A[i]$   
 7. Swap( $A[i], \text{largest}$ )  $\rightarrow$   $i, \text{largest}$   
 8. max-Heapify( $A, \text{largest}$ )  $\rightarrow$  بازگشته

$O(h) = O(\log n)$  درست زمان  
 ارسانع درست

می خواهیم اکارای دلخواه را بسی سبک نمایم.  
 شرط این است که هر دوی از  $\frac{n}{2}$  که بزرگتر از  $\text{largest}$  باشند را بزرگتر کنیم.

**Build-Max-Heap(A, n):**  $A[1| | | | \dots |n]$

1. for  $i = \lfloor \frac{n}{2} \rfloor$  down to 1 do:  
 2. max-Heapify( $A, i$ )

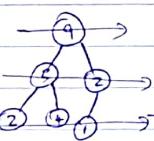
$O(n \log n) \rightarrow$   $\frac{n}{2} > \text{HeapSize}$

بین حدیث  $\leftarrow$  این که درین حالت به حدیث و رسالت پیری

کلیت من سیم زن بین اموریم بین  $n \cdot \log n$  می خوردند  $tight$   
 شدیده از قاعده همینه  
 $c \log n \log n$   
 $c \times 2 \times 2 \times \frac{n}{4}$   
 نهال NAHAL

## بازهایی Max-Heap

سطر به سطر ← از بالا به پایین ← از چپ به راست شیرهای درایه



منذریم. سلا'

منذریم.

$$\left. \begin{array}{l} \text{parent}(i) = \lfloor \frac{i}{2} \rfloor \\ \text{left}(i) = 2i \\ \text{right} = 2i + 1 \end{array} \right\}$$

- یک درخت بازیگر کامل که زیردرخت چیزی را داشت آن  
Max-Heap بگوییم.

Max-Heapify(A, i):

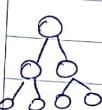
1. if  $\text{left}(i) < \text{HeapSize}[A]$  and  $A[\text{left}(i)] > A[i]$ ;
2.      $\text{largest} = \text{left}(i)$
3. else  $\text{largest} = i$
4. if  $\text{right}(i) < \text{HeapSize}[A]$  and  $A[\text{right}(i)] > A[\text{largest}]$ ,  
 $\text{largest} = \text{right}(i)$

## Root درخت

(Complete binary tree) ← مامل

دتر (دتر) در عده از برگ های پایین بعضاً برگ های (سطر آخر)

امداد راست خفت بوده ایم. مامل ← (سطر آخر)



درخت راست کامل است. نه کامل

بروزرسانی مقادیر فرآیند است، Max-Heap

بروزرسانی مقادیر فرآیند است، Min-Heap

Maximum:  $O(1)$

همه ریشه درخت است.

Minimum:  $O(n)$

نهایت پنهانی درخت

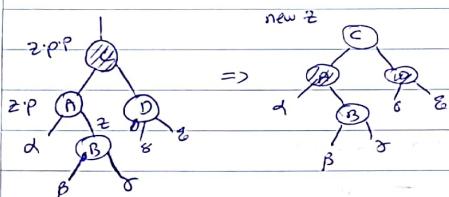
NAHAL

اینجا معنی است که رئیر باشد و ملبو<sup>۹</sup> رین هم فرجز است.

Subject.— تحریر داده (خواسته)  $T_{\text{req}} + \text{color} = \text{Block}$  می شود.

حالات پنهان (Maintenance) : غرض من نیم (Maintenance) در اینجا  $\text{b} \oplus \text{c} \oplus \text{d}$  است. این کلمه را مطلع while درست است. آب است و نیم در اینجا کام بدری نیز درست است.

$$\textcircled{2} \quad y = z \cdot p \cdot p, \text{right} \quad , \quad \text{رس ي فرمات:}$$



جدول رامز جلدی بعد  
ن ہما ز.م.ج. ح چوبی اس س و دلکش قرقرہ اس س

Subject: \_\_\_\_\_ Date: \_\_\_\_\_

با سفیدی از App Invariant ← Red-Black درج در محت عمل دارم

حالات اولیه (عمل از اجرای خدمت) (while)

درست است بیوں color = RED را درایرا خواهد داده میسوزد. (a)

درایرا درست real-black است، بیش از زیر پاسه است. (b)

بهره اضاعه همین بدرست real-black و دارن دارن color = RED (c)

ویزین ۱ و ۳ و ۵ نفی توانند تغییر مسود. اگر زیر پاسه ویزین (d)

ویزین ۲ نفی توانند تغییر مسود. بیش ویزین (e) تغییر مسود. و اگر زیر پاسه (f)

ویزین ۲ نفی توانند تغییر مسود. و ویزین ۴ همان است تغییر مسود. (g)

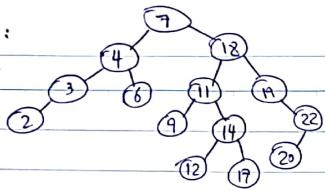
زیر پاسه قریز باشد. (h)

حالت نهایی (Termination)

while زیر پاسه color == RED do {

میزین ۴ تغییر نمی کند شکوفه بیوں زیر پاسه است. (i)

left-rotate ( $T, x$ ) :



RB\_Insert ( $T, z$ ) :

1.  $y = T.\text{null}$
2.  $x = T.\text{root}$
3. while  $x \neq T.\text{null}$  :
4.      $y = x$
5.     if  $z.\text{key} < x.\text{key}$  :
6.          $x = x.\text{left}$
7.     else  $x = x.\text{right}$
8.  $z.p = y$
9. if  $y == T.\text{null}$  then :
10.      $T.\text{root} = z$
11. if  $z.\text{key} < y.\text{key}$  :
12.      $y.\text{left} = z$
13. else  $y.\text{right} = z$

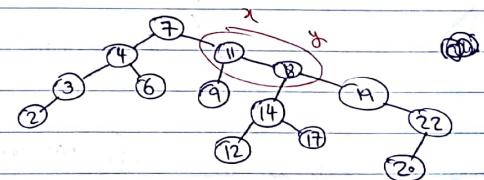
ناھال  
NAHAL

Left-Rotate ( $T, x$ ) :

- Subject \_\_\_\_\_ Date \_\_\_\_\_
1.  $y = x.\text{right}$
  2.  ~~$x.\text{right} = y.\text{left}$~~
  3. if  $y.\text{left} \neq T.\text{null}$  then
  4.      $y.\text{left}.p = x$
  5.  $y.p = x.p$
  6. if  $x.p == T.\text{null}$  then :
  7.      $T.\text{root} = y$
  8. else if  $x == x.p.\text{left}$  then
  9.      $x.p.\text{left} = y$
  10. else  $x.p.\text{right} = y$
  11.  $y.\text{left} = x$
  12.  $x.p = x$

$T(\text{Left-Rotate}) = \Theta(1)$

exp)



ناھال  
NAHAL

٩.  $\text{if } z == z \cdot p \cdot \text{right} :$

١٠. Subject  $z = z \cdot p$  } Case 2 Date: ..
١١. left\_rotate ( $T, z$ ) } Case 2
١٢.  $z \cdot p \cdot \text{color} = \text{Black}$
١٣.  $z \cdot p \cdot p \cdot \text{color} = \text{RED}$  } Case 3
١٤. Right-Rotate ( $T, z \cdot p \cdot p$ )
١٥. else ( مثل حين ملئ left بورقة من سود right )
١٦.  $T \cdot \text{root} \cdot \text{color} = \text{Black}$  → پایه روت برکت  $\rightarrow$  ( ٢ ) ( تعمیم و خوبی )

درست است . تو از هار همیشه درست بگان این سؤال :

ا) اگر  $z \cdot p \cdot \text{color} = \text{Black}$  بگزینیم . ( ١ )

ب) اگر درخت و کن هار  $\text{Red-Black}$  باشند ( ٢ ) ( رابطه بین دو نویس این است که  $z$  روت است )

را فهمی نمایند . اگر ویرگول ( ٢ ) را بزنند بین دو نویس این است که  $z$  روت است

و همچو قدرمی است . اگر ویرگول ( ٤ ) را بزنند بین دو نویس این است که  $z$  روت است

هردو قدرمیند .

\* داده زیر ایجاد کنید insert

\* موردنی است و موردنی نیست

Date: ۴-۲-۲۲

**RB - Insert - Fixup ( $T, z$ ):**

Subject: 7. while ( $z.p.color == RED$ ):

if  $z.p == z.p.p.left$ :  $z.p.p$

3.  $y = z.p.p.right \rightarrow z \leftarrow y$

if  $y.color == RED$ :

5.  $z.p.color = BLACK$

6. Case 1  $y.color = BLACK$

7.  $z.p.p.color = RED$

8.  $z = z.p.p \rightarrow$  مسئل را بروز دارد  
بالاتر این مسئله است

9. else if  $y.color == Black$  and  $z = z.p.left$ :

10.  $z.p.color = Black$

11. Case 2  $z.p.p.color = RED$

12. Right-Rotate ( $T, z.p.p$ )

13. else if  $y.color == Black$  and  $z = z.p.right$ :

14. Case 2  $z = z.p \rightarrow$  مسئله زیر نیست  
Left-Rotate ( $T, z$ )

15. Case 3  $\approx$  مسئله زیر نیست

16.  $z.p.color = Black$

17.  $z.p.p.color = RED$

18. Right-Rotate ( $T, z.p.p$ )

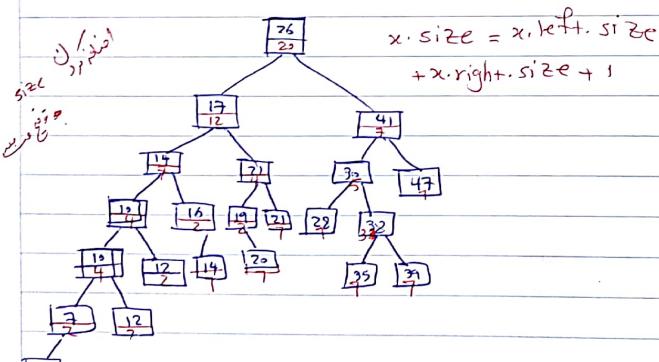
Subject:	Date:
Ranking	سل میست
OS_Rank ( $T, x$ )	نحوی به دادن $x$ در $T$ که دارای $i$ اندیخته باشد.
1. $r = x.left.size + 1$	کل داشت از نوع درخت را
2. $y = x$	$O(n \log n)$ نمایش
3. while $y \neq T.root$ do:	است.
4. if $y = y.p.right$ :	
5. $r = r + y.left.size + 1$	
6. $y = y.p$	
7. return $r$	
Fixup و ترتیب:	
بر عمل درج: خوازه در درج برد	نحوی در درج در درج برد
و بعد از آن از دادن تا زیر جایی که دارای $i$ اندیخته باشد	جهت دادن از دادن در درج برد
و بعد از آن عمل rotate این جهت	جهت دادن در درج برد
و بعد از آن size را تغییر داد	جهت دادن در درج برد
OS	نوار

Subject:	Date:
برای دادن $x$ در $T$ که دارای $i$ اندیخته باشد.	برای دادن $x$ در $T$ که دارای $i$ اندیخته باشد.
OS_select ( $x, k$ )	دریک روش برای دادن $x$ در $T$ که دارای $i$ اندیخته باشد.
1. $r = x.left.size + 1$	
2. if $k == r$ :	
3. return $x$	
4. else if $k < r$ : return OS-select ( $x.left, k$ )	برای دادن $x$ در $T$ که دارای $i$ اندیخته باشد
5. else: return OS-select ( $x.right, k - r$ )	نادر و اعجوب باید در سمت راست
	نهایل ( $k=1$ ) اصلی خود
	پس از
	کل از دادن در درج برد
	هدان خود $-k$ که
	راحاب می شوند.

## Augmenting Data Structures

ل ہمیں سکو احوالات بے سختی دادھا موجور اضافہ میں نہیں تھے جو انہیں مستثنی  
ل پیش کی را جا سکتی دادھل شد.

نکته: جستجو و نتایج کو بررسی کردن از نتایج جستجو که نتایج جستجو کردن از نتایج بررسی کردن می‌باشد.



الخطوة الأولى: إدخال الكلمات المفتاحية في نافذة "Insert" في Microsoft Word.

if  $w \cdot \text{color} == \text{RED}$  :

Subject: \_\_\_\_\_ W. COLOR = BLACK Date: \_\_\_\_\_

x.p.color = RED

left-rotate ( $T, x, p$ )

$$\omega = x \cdot p \cdot \text{right}$$

if ~~w.left~~.color == BLACK and w.right.

$\omega \cdot \text{color} = \text{REO}$

$$x = x \cdot p$$

else  
if w.right.color == BLACK :

w.left.color = BLACK

ω · Colr = BFD

right-rotate ( $T; w$ )

Wavelength =  $\lambda$ : color

~~2.2 : color = BLACK~~

w-right-color = BLACK

left  $\rightarrow$   $(\pi \nu, e)$

else (same as then ... left  $\leftrightarrow$  right)

X-MAT - BLACK

if  $y.p == z:$

$x.p = y$

else:

RB-Transplant( $T, y, z.right$ )

$y.right = z.right$

$y.right.p = y$

RB-Transplant( $T, z, y$ )

$y.left = z.left$

$y.left.p = y$

$y.color = z.color$

if  $y.original_color == \text{BLACK}:$

RB\_Delete-Fixup( $T, x$ )

یعنی مکانی که  $y$  باید باشد،  $y.original_color$  است (یادشود)

برای پاپ و غیرز قرمز کن جایشین آن سور

ترفته شده است  $x.p$  و  $x$  هم هم

RB\_Delete-Fixup( $T, x$ ):

while  $x \neq T.root$  and  $x.color == \text{BLACK}:$

if  $x == x.p.left:$

$w = x.p.right$

NAHAL

Subject:

RED-BLACK ترقیت در حفظ درخت

RB-Transplant( $T, u, v$ ):

if  $u.p == T.null:$

$T.root = v$

elif  $u == u.p.left:$

$u.p.left = v$

else:  ~~$u$~~   $u.p.right = v$

$v.p = u.p$

RB-delete( $T, z$ ):

$y = z$

$y.original_color = y.color$

if  $z.left == T.null:$

RB-Transplant( $T, z, z.left$ )

elif  $z.right == T.null:$

RB-Transplant( $T, z, z.right$ )

else:

$y = Tree-Min(z.right)$

$y.original_color = y.color$

$x = y.right$

NAHAL

### Interval Tree

$\forall x \in R : t_1 \leq x \leq t_2 \Rightarrow i \in \text{Interval}$

$i.\text{low} = t_1, i.\text{high} = t_2$

若有交集 (overlap)  $i.\text{low} \leq t_1 \leq i.\text{high} \leq t_2$

$i.\text{high} > i.\text{low}$  不重叠 (non-overlap)

$(i.\text{low} > i.\text{high})$  为空 (empty)

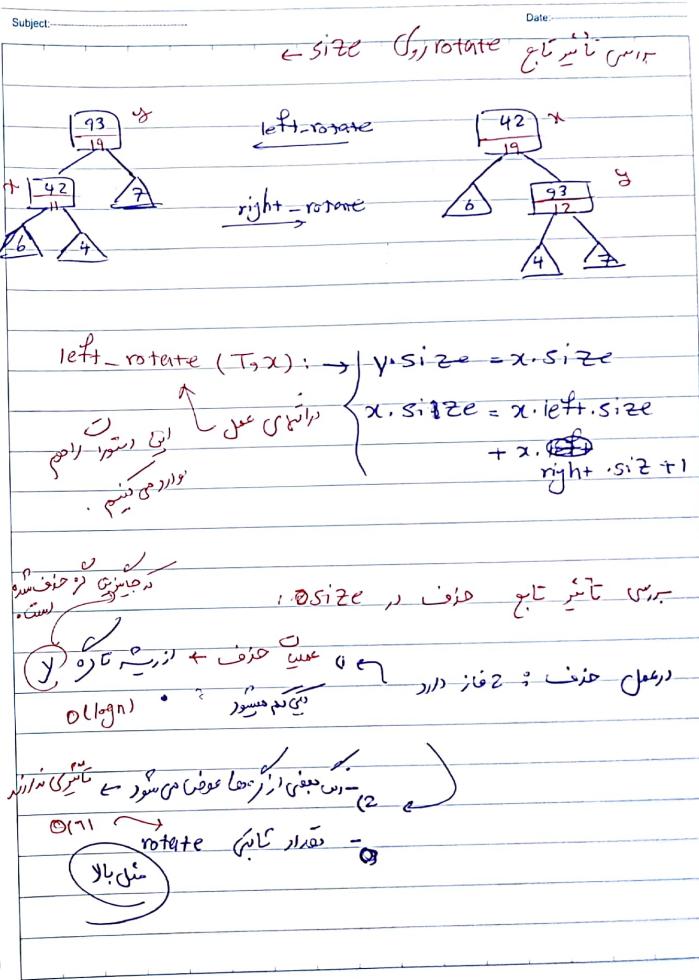
若  $x$  在  $t_1$  和  $t_2$  之间，则  $x$  在  $i$  中，即  $i$  包含  $x$  (包含 (contains))

若  $x$  在  $i$  的左右边界之外，则  $x$  不在  $i$  中 (not contains)

Interval-Delete( $T, x$ ) ② Interval-Insert( $T, x$ ) ①

Interval-Search( $T, i$ ) ③

插入操作 (Insert Operation)



Interval-search( $T, i$ ):  $O(n \lg n)$  : الـ  $\Delta$  الـ  $\Delta$

Subject

1.  $x = T \cdot r^{0.5}$

Date:-

2. while ( $x + T$  is null and does not overlap  $x$ .int do:

3. if  $x.\text{left\_}! = T.\text{null}$  end  ~~$T.\text{null}$~~   $x.\text{left\_max}, i.\text{low}, i.\text{high}$

$x = x.\text{left}$

else  $z = x.\text{right}$

• return  $x$

\* و نحوه اصلاح  $\rightarrow \max$   $O(\log n)$   $\leftarrow$  size ( $\Omega(n^2)$ )

اُسیات درستی کے سوریم کے search کو

نمایت حلقه  $\rightarrow$  ارورت  $T_{\text{سکل}} \leftarrow$  بزرگ نماین  $\rightarrow$  این پیشنهاد دارد

نهاده زیردرفت (نام) بازیز سهل چشم بازدید است.

درستگار حلقه while مستمر و Initialization ( خانه پرینت ) :

•  $\omega$  < T. most

درایی هات ثابت حلق بی علاوه به برقرار است. هنون آنها

• سے ۱۲ نویں

$$x \cdot \text{int} \cdot \text{bw} = x \cdot \text{key}$$

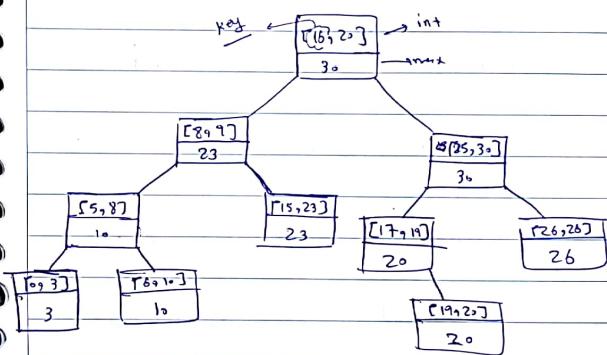
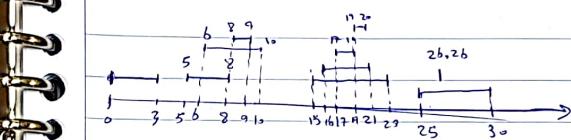
**Subject:** \_\_\_\_\_

کسیر هر دو پاسی بارہ است.

Date: \_\_\_\_\_



Page 5 of 6



$x.\max = \max\{x.\text{int\_high}, x.\text{left\_max}, x.\text{right\_max}\}$

$\vdash (\cup_{i \in I} s_i) \leftarrow$  maintenance

**Subject:-**

Date:-

Subject: \_\_\_\_\_ Date: \_\_\_\_\_  
اگر رایجہار میں کام صدیع، نائب حلم (رست است) ۰ ۱۷۶۴ء۔ دلیل

**ڪام**  **بعدك تئي** ڪائبِ ڏلَم درس ٻالٽ.

8. ~~(left)~~ low  $\leftarrow$  1.0m

اگر مسأله else کے ساتھ اپنے جواب ملے تو اس کو اجرا سکوں (خط ۵)

i<sup>ː</sup>. high { i'. high

$\leftarrow i.\text{leftFollow} \leftarrow i.\text{low}$

time high چوں

نیجہ میں سرحدیم → اُبادتیں ہیں استاراں ھندوار،

برقرار باس (خواه) if اگر سئانیم و می‌شوند باید حمل

برقراری است : (رایج)  $\leftarrow$   $i \cdot \text{high}$   $\leftarrow x \cdot \text{left} \cdot \text{max}$

اگر اُزو اسٹریک نئیتہ بائے متھ حل دست و میرنہ بائے

برای هر درز درخت راست  $i$ .high  $i'$ .low

معنی ؟ بایزهار سوت راست استاد نادر.