

مراجع دليل:

ساختار داده:

① introduction to Algorithms C. L. R. S 3rd ed

② درس قدرتی → نویسنده داده و الگوریتم ها

۵ انتشار بین المللی

۱۲ بیانیه

۳ ترجمه کریم زرده

الگوریتم ۲

ساختار داده: گنجی از حافظه در میان الگوریتم را ذخیره نموده



A₁ درست



کلیل نظر الگوریتم ۲

خطان = عدالت

1. $x = a$

2. for i := 0 to n do:

3. $x = x + r$

محضیه

b c₁

c₂ c₃

c₄ c₅

c₆ c₇

c₈ c₉

c₁₀ c₁₁

c₁₂ c₁₃

c₁₄ c₁₅

c₁₆ c₁₇

c₁₈ c₁₉

c₂₀ c₂₁

c₂₂ c₂₃

c₂₄ c₂₅

c₂₆ c₂₇

c₂₈ c₂₉

c₃₀ c₃₁

c₃₂ c₃₃

c₃₄ c₃₅

c₃₆ c₃₇

c₃₈ c₃₉

c₄₀ c₄₁

c₄₂ c₄₃

c₄₄ c₄₅

c₄₆ c₄₇

c₄₈ c₄₉

c₅₀ c₅₁

c₅₂ c₅₃

c₅₄ c₅₅

c₅₆ c₅₇

c₅₈ c₅₉

c₆₀ c₆₁

c₆₂ c₆₃

c₆₄ c₆₅

c₆₆ c₆₇

c₆₈ c₆₉

c₇₀ c₇₁

c₇₂ c₇₃

c₇₄ c₇₅

c₇₆ c₇₇

c₇₈ c₇₉

c₈₀ c₈₁

c₈₂ c₈₃

c₈₄ c₈₅

c₈₆ c₈₇

c₈₈ c₈₉

c₉₀ c₉₁

c₉₂ c₉₃

c₉₄ c₉₅

c₉₆ c₉₇

c₉₈ c₉₉

c₁₀₀ c₁₀₁

c₁₀₂ c₁₀₃

c₁₀₄ c₁₀₅

c₁₀₆ c₁₀₇

c₁₀₈ c₁₀₉

c₁₁₀ c₁₁₁

c₁₁₂ c₁₁₃

c₁₁₄ c₁₁₅

c₁₁₆ c₁₁₇

c₁₁₈ c₁₁₉

c₁₂₀ c₁₂₁

c₁₂₂ c₁₂₃

c₁₂₄ c₁₂₅

c₁₂₆ c₁₂₇

c₁₂₈ c₁₂₉

c₁₃₀ c₁₃₁

c₁₃₂ c₁₃₃

c₁₃₄ c₁₃₅

c₁₃₆ c₁₃₇

c₁₃₈ c₁₃₉

c₁₄₀ c₁₄₁

c₁₄₂ c₁₄₃

c₁₄₄ c₁₄₅

c₁₄₆ c₁₄₇

c₁₄₈ c₁₄₉

c₁₅₀ c₁₅₁

c₁₅₂ c₁₅₃

c₁₅₄ c₁₅₅

c₁₅₆ c₁₅₇

c₁₅₈ c₁₅₉

c₁₆₀ c₁₆₁

c₁₆₂ c₁₆₃

c₁₆₄ c₁₆₅

c₁₆₆ c₁₆₇

c₁₆₈ c₁₆₉

c₁₇₀ c₁₇₁

c₁₇₂ c₁₇₃

c₁₇₄ c₁₇₅

c₁₇₆ c₁₇₇

c₁₇₈ c₁₇₉

c₁₈₀ c₁₈₁

c₁₈₂ c₁₈₃

c₁₈₄ c₁₈₅

c₁₈₆ c₁₈₇

c₁₈₈ c₁₈₉

c₁₉₀ c₁₉₁

c₁₉₂ c₁₉₃

c₁₉₄ c₁₉₅

c₁₉₆ c₁₉₇

c₁₉₈ c₁₉₉

c₂₀₀ c₂₀₁

c₂₀₂ c₂₀₃

c₂₀₄ c₂₀₅

c₂₀₆ c₂₀₇

c₂₀₈ c₂₀₉

c₂₁₀ c₂₁₁

c₂₁₂ c₂₁₃

c₂₁₄ c₂₁₅

c₂₁₆ c₂₁₇

c₂₁₈ c₂₁₉

c₂₂₀ c₂₂₁

c₂₂₂ c₂₂₃

c₂₂₄ c₂₂₅

c₂₂₆ c₂₂₇

c₂₂₈ c₂₂₉

c₂₃₀ c₂₃₁

c₂₃₂ c₂₃₃

c₂₃₄ c₂₃₅

c₂₃₆ c₂₃₇

c₂₃₈ c₂₃₉

c₂₄₀ c₂₄₁

c₂₄₂ c₂₄₃

c₂₄₄ c₂₄₅

c₂₄₆ c₂₄₇

c₂₄₈ c₂₄₉

c₂₅₀ c₂₅₁

c₂₅₂ c₂₅₃

c₂₅₄ c₂₅₅

c₂₅₆ c₂₅₇

c₂₅₈ c₂₅₉

c₂₆₀ c₂₆₁

c₂₆₂ c₂₆₃

c₂₆₄ c₂₆₅

c₂₆₆ c₂₆₇

c₂₆₈ c₂₆₉

c₂₇₀ c₂₇₁

c₂₇₂ c₂₇₃

c₂₇₄ c₂₇₅

c₂₇₆ c₂₇₇

c₂₇₈ c₂₇₉

c₂₈₀ c₂₈₁

c₂₈₂ c₂₈₃

c₂₈₄ c₂₈₅

c₂₈₆ c₂₈₇

c₂₈₈ c₂₈₉

c₂₉₀ c₂₉₁

c₂₉₂ c₂₉₃

c₂₉₄ c₂₉₅

c₂₉₆ c₂₉₇

c₂₉₈ c₂₉₉

c₃₀₀ c₃₀₁

c₃₀₂ c₃₀₃

c₃₀₄ c₃₀₅

c₃₀₆ c₃₀₇

c₃₀₈ c₃₀₉

c₃₁₀ c₃₁₁

c₃₁₂ c₃₁₃

c₃₁₄ c₃₁₅

c₃₁₆ c₃₁₇

c₃₁₈ c₃₁₉

c₃₂₀ c₃₂₁

c₃₂₂ c₃₂₃

c₃₂₄ c₃₂₅

c₃₂₆ c₃₂₇

c₃₂₈ c₃₂₉

c₃₃₀ c₃₃₁

c₃₃₂ c₃₃₃

c₃₃₄ c₃₃₅

c₃₃₆ c₃₃₇

c₃₃₈ c₃₃₉

c₃₄₀ c₃₄₁

c₃₄₂ c₃₄₃

c₃₄₄ c₃₄₅

c₃₄₆ c₃₄₇

c₃₄₈ c₃₄₉

c₃₅₀ c₃₅₁

c₃₅₂ c₃₅₃

c₃₅₄ c₃₅₅

c₃₅₆ c₃₅₇

c₃₅₈ c₃₅₉

c₃₆₀ c₃₆₁

c₃₆₂ c₃₆₃

c₃₆₄ c₃₆₅

c₃₆₆ c₃₆₇

c₃₆₈ c₃₆₉

c₃₇₀ c₃₇₁

c₃₇₂ c₃₇₃

c₃₇₄ c₃₇₅

c₃₇₆ c₃₇₇

c₃₇₈ c₃₇₉

c₃₈₀ c₃₈₁

c₃₈₂ c₃₈₃

c₃₈₄ c₃₈₅

c₃₈₆ c₃₈₇

c₃₈₈ c₃₈₉

c₃₉₀ c₃₉₁

c₃₉₂ c₃₉₃

c_{394</sub}

Subject _____
Date _____

خالل رتب سازی دفعی : insertion sort

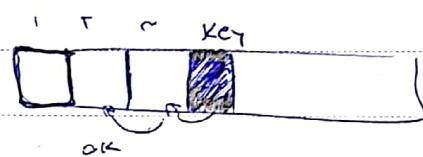
دستور اجرایی: آرایه A و عدد کوچکی n

(فرانسی) مرتب سازی دفعی (اسعاد الدین زریش) (فرانسی)

Incremental

insertion sort: 1. for $j \leftarrow 1$ to n do \rightarrow

2. $key \leftarrow A[j]$



3. $i \leftarrow j - 1$

while $key < A[i]$ and $i > 0$ do

0. $A[i+1] \leftarrow A[i]$

1. $i \leftarrow i - 1$

2. $A[i+1] \leftarrow key$

3. end while

| | 1 | $n-1$ | C_1 |
|-------|-----------|-------|-------|
| i | $n-1$ | C_r | |
| r | $n-1$ | C_a | |
| t_j | $t_j + 1$ | C_f | |
| d | t_j | C_d | |
| j | t_j | C_u | |
| v | $n-1$ | C_v | |

برای جایگزینی t_j با t_{j+1} در حلقه while

برای جایگزینی t_j با t_{j+1} در حلقه for

Subject

Date

$$T(n) = C_1 n + C_r (n-1) + C_c (n-1) + C_e \sum_{j=r}^n (t_j + 1)$$

$$+ C_d \sum_{j=r}^n t_j + C_g \sum_{j=r}^n t_j + C_v (n-1)$$

$$\therefore t_j < j-1$$

$$T_n = C_1 n + C_r (n-1) + C_c (n-1) + C_e (n-1) + C_v (n-1) \quad (t_j = j-1) \quad \text{worst case}$$
$$= an + b \quad \text{میکوئی مسیری باری}$$

$$T_n = C_1 n + C_r (n-1) + C_c (n-1) + C_e \sum_{j=r}^n (j-1) \quad : \text{Average Case}$$

$$+ C_d \sum_{j=r}^n (j-1) + C_g \sum_{j=r}^n (j-1) + C_v (n-1)$$

$$= C_1 n + C_r (n-1) + C_c (n-1) + C_e \times \frac{(n-1)(n+r)}{r} + C_d \frac{(n-1)(1+n-1)}{r}$$

$$C_d \frac{(n-1)(1+n-1)}{r} + C_v (n-1) = a'n + b \cdot n + c \quad \text{رسانی}$$

$$t_j = \frac{j-1}{r} \quad : (\text{Average Case}) \quad \text{رسانی}$$

$$T_n = an + bn + c$$

Bubble sort

رسانی

Subject _____
Date _____

bubble-sort ($A_{0,n}$)

1. for $i \leftarrow n-1$ down to 1 do

2. for $j = 1$ to i do

3. if ($A[j] > A[j+1]$) {

4. Swap ($A[j], A[j+1]$)

}

$$\begin{aligned} T(n) &\leq C_1 \cdot n + C_r \sum_{i=1}^{n-1} (i+1) + C_w \sum_{i=1}^{n-1} i \\ &= C_1 \cdot n + \frac{n-1}{2} \cdot (n-1)C_r + C_w \frac{(n-1)n}{2} \\ &= an^2 + bn + c \end{aligned}$$

الخطوة i تتم بـ i مرات . كل مررت يتم في خطوة i مرات .

مقدار الخطوة i متساوية

$\omega, o, \theta, \Omega, O$ سطح

little omega

little theta

teta

big omega

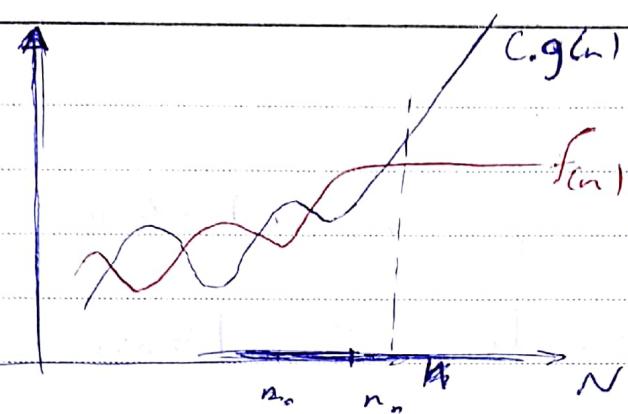
big O

$f(n) \in O(g(n)) \Leftrightarrow \exists c > 0, \exists n_0 > 0 \mid \forall n > n_0 \quad f(n) \leq C \cdot g(n)$

$g(n)$ يمثل سطح $f(n)$ في

Subject

Date



(C.g(n)) علیه (f(n))

(C.g(n)) علیه (f(n))

$$\text{برای } f(n) : \lim_{n \rightarrow \infty} \frac{g(n)}{f(n)} = C \neq 0 \text{ میشود}$$

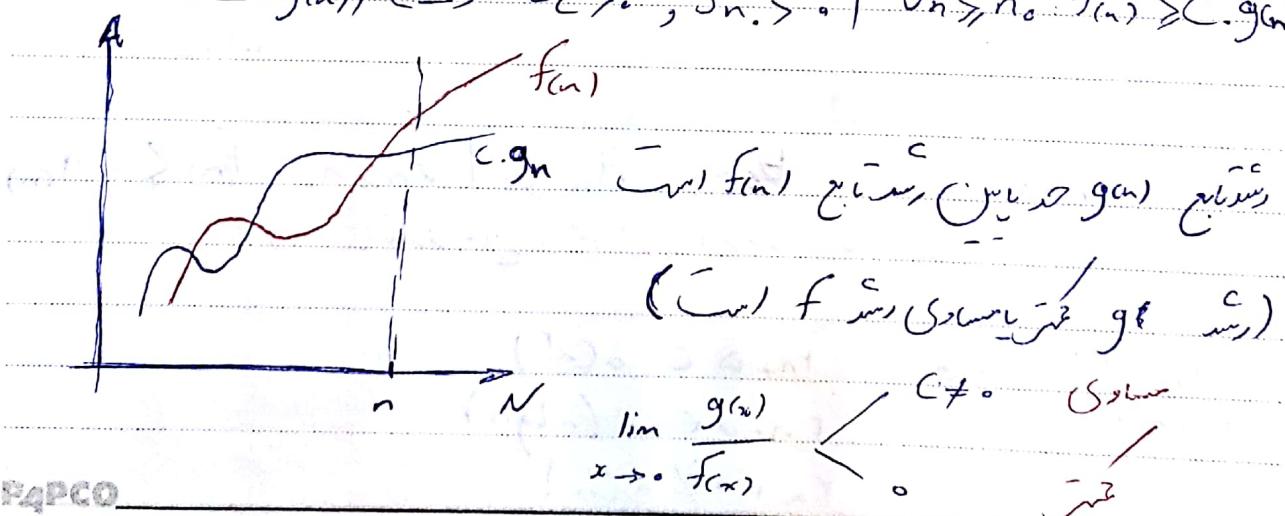
$$ج) \lim_{n \rightarrow \infty} \frac{r_n}{n} = \infty \quad n \in O(r_n) \quad n \notin O(\sqrt{n})$$

$$\lim_{n \rightarrow \infty} \frac{r_n}{n} = r \quad n \in O(r_n)$$

برای big O \leftarrow $\exists c > 0, \exists n_0 | \forall n \geq n_0, f(n) \geq c.g(n)$

big omega: Ω

$$f(n) \in \Omega(g(n)) \Leftrightarrow \exists c > 0, \exists n_0 | \forall n \geq n_0, f(n) \geq c.g(n)$$



$$\lim_{x \rightarrow \infty} \frac{g(x)}{f(x)} = C \neq 0 \text{ میشود}$$

R&PCO

$$\textcircled{2} \quad n \in \Omega(r_n), \lim_{n \rightarrow \infty} \frac{r_n}{n} = r \quad \lim_{n \rightarrow \infty} \frac{n}{\sqrt{n}} = \infty \quad n \in \Omega(\sqrt{n})$$

Subject
Date

$$n \in \Omega(n^r)$$

(جَعَلَهُمْ جَنَاحَاتٍ) : teta Θ

$$f(n) \in \mathcal{O}(g(n)) \iff \exists c_1 > 0, \exists c_r > 0, \exists n_0 > 0 \mid \forall n \geq n_0$$

$$c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$$

$$n \in \theta(p_n), p_n \in \theta(n), n \in \theta(\sqrt{n}), n \in \theta(n^r)$$

(١٣) $f_{(n)} + O(f_{(n)}) \in \theta(f_{(n)})$ ✓ $\text{تم}\text{تبر}$

i) $f \in O(g(n))$

فرزی نهاد در حکایتی از ۳۷ حالت برج و بسیار

④ $f \in \Omega(g(n))$

$f \in \Theta(g(n))$

$$f = n^{1 + \sin x} \quad g = n^{\cos x}$$

little o : o

$f(n) \in o(g(n)) \iff \exists c > 0 \mid \forall n \geq n_0 \quad f(n) < c \cdot g(n)$

که دلیل این خواستگاری ممکن است در تصور عکسی کردن از آن است

$$\lim_{n \rightarrow \infty} \frac{g(n)}{f(n)} = \infty \quad r_{n+2} \in o(n^r) \\ r_{n+2} \in o(n \lg n)$$

$\tau = \beta t + (-)$

PAPCO

$$r_n + o \notin a(n)$$

$$r_n + \omega \in o(\sqrt{n})$$

Subject _____

Date _____

$$\log_b^n = \log_a^n \times \log_b^a \quad \text{বিকল } \rightarrow \log_a^n \in \Theta \log_b^n$$
$$a^n \in o(b^n)$$

$$n^{\log_a(n)} \in o(n^{\log_a(n)}) \quad \text{বিকল } \rightarrow \text{বিকল } \quad *$$

$a > 1$

$$(\log n)^a ? n^c \rightarrow (\log n)^a \in o(n^c) \quad *$$

little omega - w

$$f(n) \in w(g(n)) \Leftrightarrow \exists c > 0 \exists n_0 \mid \forall n > n_0, f(n) > c.g(n)$$

$$\lim_{n \rightarrow \infty} \frac{g(n)}{f(n)} = 0$$

$$r_{n+o} \in w(r_n)$$

$$r_{n+o} \notin w(n) \quad r_{n+o} \notin (n)$$

$$f \in O(g) \& g \in O(h) \Rightarrow f \in O(h) \rightarrow \text{বিকল } \text{বিকল } \dots$$

Rapco _____

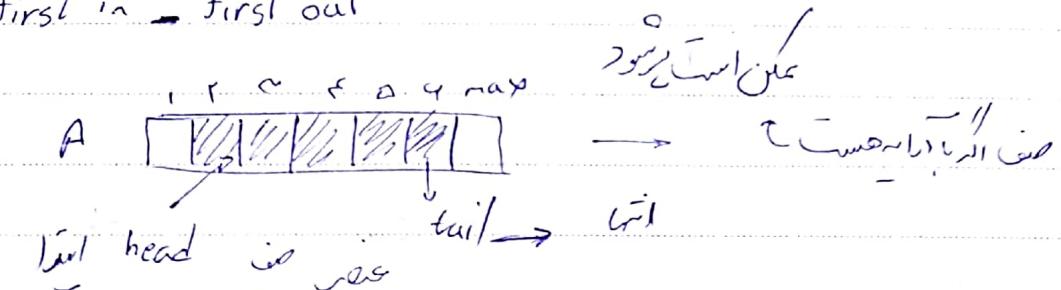
(V)

Subject _____
Date _____

FIFO

first in - first out

Queue : لائحة



Add(Q, x)

1. if $\text{tail} == \max$ then

r- error("Queue is full!")

c- return

t- else

a- $\text{tail} = \text{tail} + 1$

g- $Q[\text{tail}] = x$

Del(Q)

1. if $\text{head} > \text{tail}$ then

r- error:-

p- return

f- else

a- $x = Q[\text{head}]$

g- $\text{head} = \text{head} + 1$

v- return x:

Subject _____
Date _____

Find(θ , x)

$O(n)$ vs $O(1)$

1. For $i \leftarrow \text{head}$ to tail do
 if $a[i] == x$ then
 return true
2. return false

is Empty(θ)

$O(1)$

1. if $\text{head} > \text{tail}$ then
 return true
2. else return false

is Full

$O(1)$

1. if $\text{tail} == \text{max}$ then
 return true

2. else return false

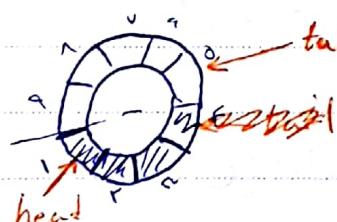
Head =

* $\text{tail} = \text{head} = 1$

tail =

tail + 1 = head

: جدی



tail : tail = head

Add: $Q[\text{tail}] = x$, tail++

Del: $x = Q[\text{head}]$, head++

is Empty(Q)

$O(1)$

if $\text{tail} == \text{head}$ then

1. $t = \text{tail} + 1$;

return true

2. if $t > \text{max}$ then

else return false

$t = 1$

$O(1)$

* ; F $t == \text{head}$ then
 return true
else return false

Subject _____
Date _____

Add(Q, x)

1. if IsFull(Q) then
 r- error
 c- return
2. Q[tail] = x
3. tail = tail + 1
4. if tail > max then
 r- tail = 1

O(n)

Del(Q)

1. if isEmpty(Q) then
 r- error
 c- return
2. x = Q[head]
3. head = head + 1
4. if head > max then
 r- head = 0

O(1)

Find(Q, x)

1. i = head
2. while i != tail do
 c- if Q[i] == x then
 c- return true
3. i = i + 1
4. if i > max then
 r- i = 1
5. return false

O(n)

a- return x

stack

new job joker

LIFO



Last in, First out.

push → stack goes in
inflow

جواب دیگر

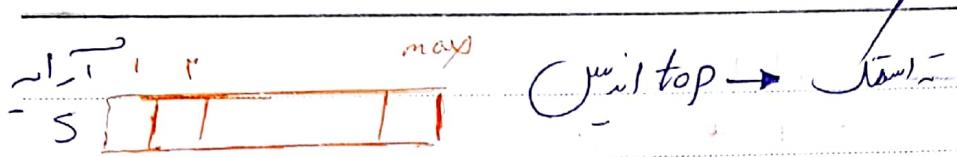
Find, IsFull, IsEmpty, Top

pop → exits
outflow

روجایی از stack goes out

Subject

Date



push: $top++$ $s[top] = x$

pop: $x = s[top]$, $top--$

top = max

is Empty (s). $O(1)$

1. if $top == 0$

r. return true

c. else return false

Push (s, x)

$O(1)$

1. if isFull(s) then

r. error

c. return

2. $top = top + 1$

3. $s[top] = x$

is Full (s). $O(1)$

1. if $top == \text{max}$

r. return true

c. else return false

Pop (s)

$O(1)$

1. if isEmpty (s) then

r. error

c. return

2. $x = s[top]$

3. $top = top - 1$

4. return x

for C language

a+b infix left associative infix prefix

a b+ right associative post fix

+ a b left associative prefix

Subject _____
Date _____

$a+b*c$

$(a+b)*c$

$a b + c * \rightarrow$ (Simpl)

$a + (b * c)$

$a b c * + \rightarrow$ (Simpl)

Postfix \Leftarrow Prefix \Leftarrow infix

taken

Postfix \Leftarrow infix

$a - b * c / d + (e - f) * g$

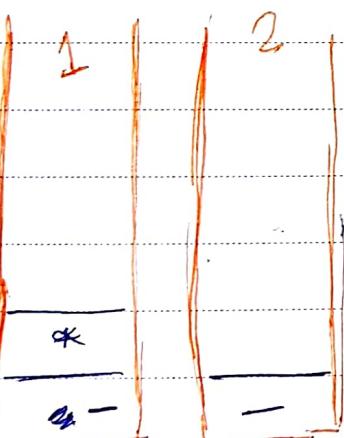
infix

$a b c * d / - e f - g * +$

infix



برای تجزیه عبارت ها استفاده کنیم stack \Leftarrow (دستورات)



خواندن سه عبارت در خروجی نسبتی سود taken

خواندن سه عبارت بازدید کوتاه بازی token

سینه در آن push می کند احری عبارت در

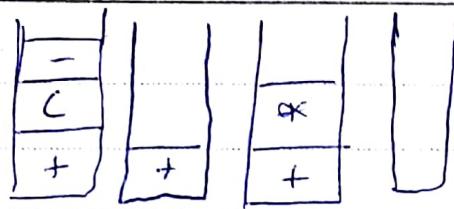


از سینه ادوات بیشتری ندارد آن باید عبارتی شود

و در خروجی نسبتی سود pop

Subject _____

Date _____

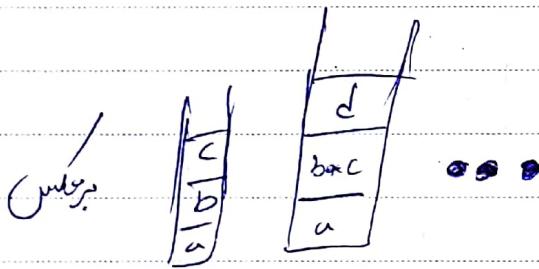


برانتر باز همراه push خواهد بود. برانتر سه قریب
است اما برانتر باز برای pop شد.

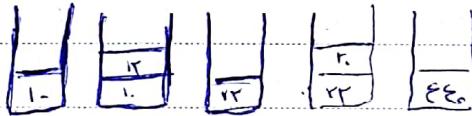
و در حقیقت اینجا ~~جی توکن~~ نویسندگی خواهد

بای لرزیایی خود (ستاد) stack باز است Postfix عبارت

بای علوفه stack را در حقیقت



(a) $1 \cdot 11 + 1 \cdot *$

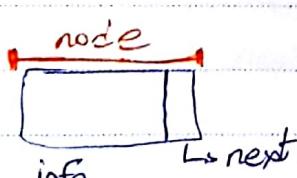
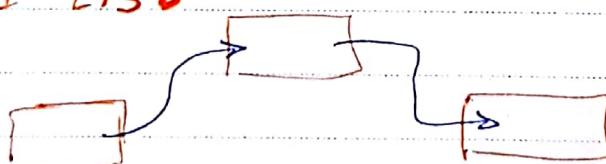


$((\alpha+\beta) - (\gamma * \delta))$

بررسی دستی برانترها



Linked List



Subject _____
Date _____

```
struct Node {
```

```
    int info;
```

```
    Node *next;
```

```
}
```

أول عقدة في القائمة

first → أول عقدة في القائمة



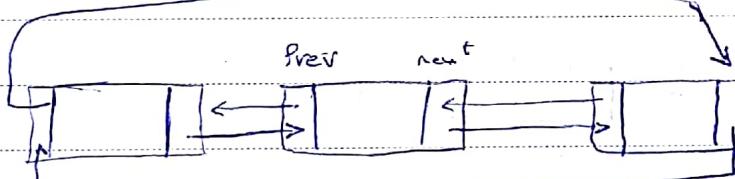
first

آخر عقدة في القائمة



first

آخر عقدة في القائمة



Node * getNode() ↗ أخذ العقدة من القائمة

```
{ Node *p;
```

```
p = (Node*) malloc (size of (struct Node));
```

```
return p;
```

```
}
```

```
void main()
```

(A)

```
{ Node * Ptr;
```

```
((*Ptr).info = 1); Ptr → info = 1;
```

```
ptr → next = null;
```

P4PCO

Subject _____
Date _____

Node * p = first;

او میکسی ایمیل

Find & O

while (P)
{ printf("info=%d", p->info);
P = P->next;

}

first

new node to copy

P →

Node * P = getNode();
P->info = 0
P->next = first;
first = P;

if info == 11 // if original node to change

Node * q = first

while (q)

{ if (q->info == 11)

Break;

q = q->next

}

if (q) { P->next = q->next;
q->next = P;

}

PAPCO

حذف اولین خودنمایی

: Obj

first



Node p = first

if (first)

{ first = first -> next

free (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)

q = P->next

P->next = P->next->next;

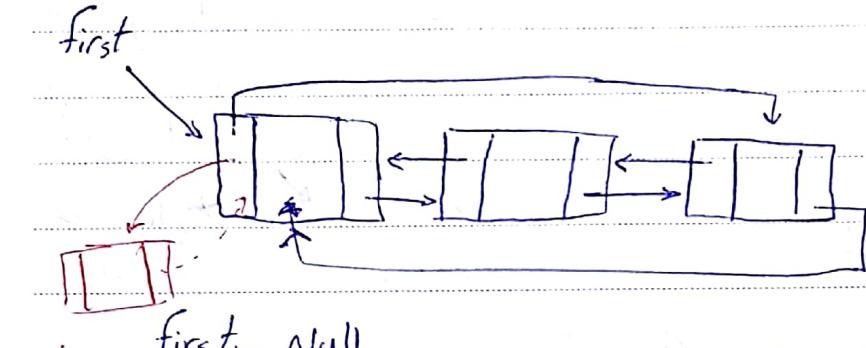
free (q);

}

P = P->next

}

لیست یوں کی در طرفہ جلوی :



Add خارج

Find(first, x) O(n)

1. if first == Null then
2. return false
3. P=first
4. do {
5. if p.info == x then
6. return true
7. P=P.next
8. while p != first
9. return false

Node *P = get Node

P.info = 0

Add(first, P)

1. if first == null
2. first = P
3. P.next = P
4. P.prev = P
5. else
6. P.next = first
7. first.prev.next = P
8. P.prev = first.prev
9. first.prev = P
10. first = P

Del(first, x)

1. P = Find(first, x)
2. if P == null
3. P.prev.next = P.next
4. P.next.prev = P.prev
5. ReFree(P)

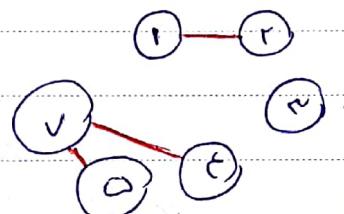
ساختار داده درخت

$$V = \{1, 2, 3, 4, 5\}$$

$$E = \{(1, 2), (2, 5), (1, 3), (3, 4), (4, 5)\}$$

$$E \subseteq V \times V$$

گراف $G = (V, E)$
 گروهی یا لیست
 vertex edge



گراف چند گزینه از هر رأس به هر اس دیگر یک سری توان نیست / خوش بیند

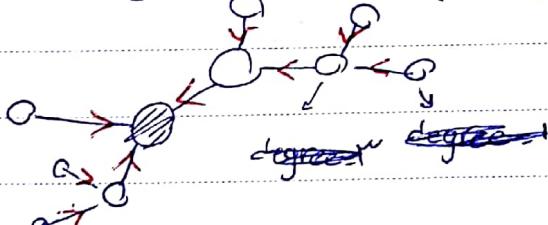
گراف بین گزینه
 گراف دار گزینه



درخت یا گراف چند گزینه است که قادر در راس است

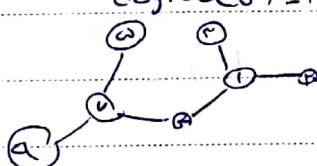
درخت دیگر: درخت دیگر را سفید گوییم و درخت دار را سبز (از رمه های دیگر راست)

نهایی یا root درخت را می بینیم



درجی راس: Node degree

$$\text{degree}(v) =$$



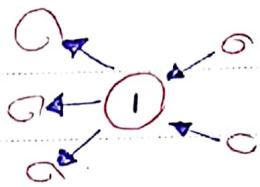
اگر حساب کن

Subject

Date

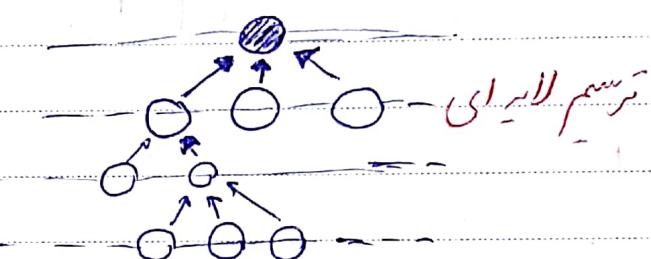
indegree = دخلی درجه (سویه)

outdegree = خارجی درجه (سویه)



$\forall v \in V$ indegree(v) ≤ 1 / ~~in and out degrees~~ / \Rightarrow Rooted tree / اسے باسم ریوتڈ تری کہا جائے گا

بعد ادھل کی تعداد کی تعداد

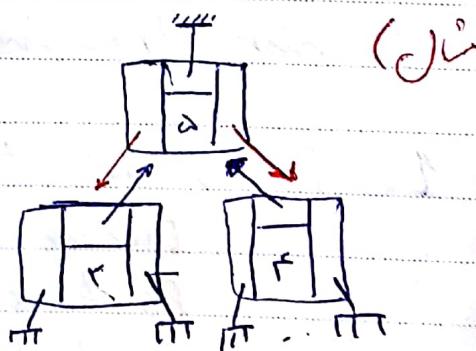
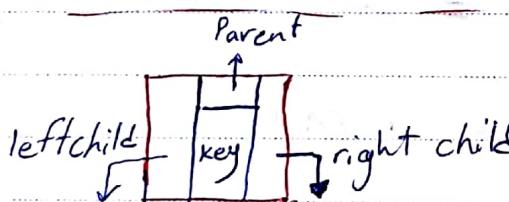


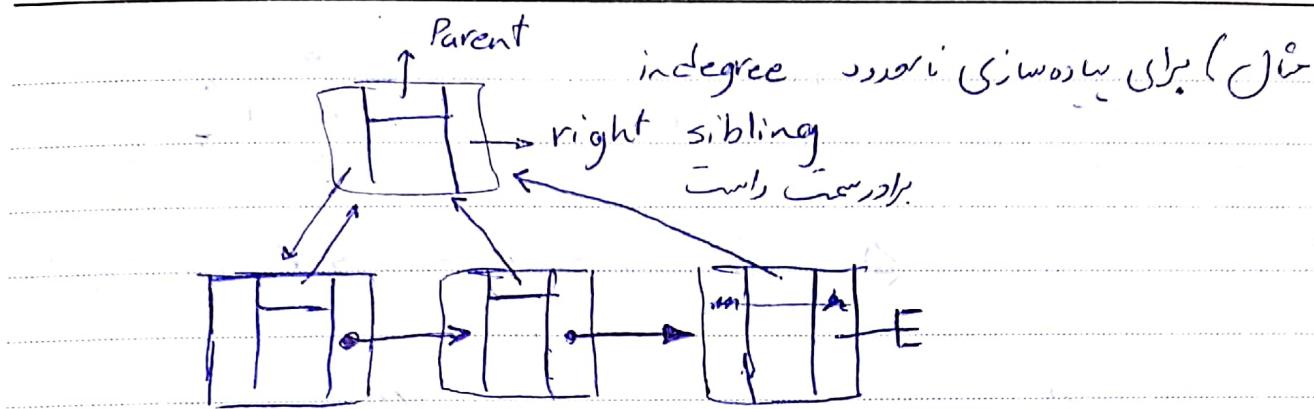
$h(v)$ ماکسیمیم گاہ از لین (تاریخ) : (height) ارتفاع کو

عوچہ کو : (depth) ناصلہ

ارتفاع درخت : ارتفاع رسی

جو ای جو دوسری گاہ کو مل جائے تو اس کو جو دوسری گاہ کو

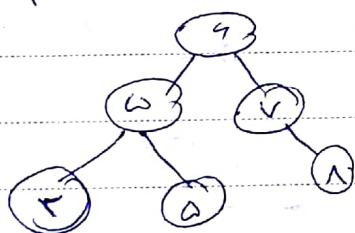




Binary search tree (BST):

دیکشنری میانی داریم که در آن دو دستور از درج و حذف است.

در جایی که $y.key < x.key$ میزد



$y.key \geq x.key$

لابیوگرامی ترتیبی = ترتیب ایندر (Inorder)

Inorder-Tree-walk(x) $O(n)$

1. if $x \neq \text{Null}$ then

 2. Inorder-Tree-walk($x.\text{left}$)

 3. Print $x.\text{key}$

 4. Inorder-Tree-walk($x.\text{right}$)

Subject _____
Date _____

$$T_n = O(n)$$

لما $T_0 = C$

فقط

$$T(n) \leq d + T(k) + T(n-k-1)$$

if $i=j$

$$T_n \leq (C+d)n + C$$

$$T_n \leq d + T(k) + T(n-k-1) \leq d + [(C+d)k + C] + [(C+d)(n-k-1) + C]$$

$$\leq (C+d)n + C$$

الآن سأكتب في

Tree-search(x, k)

$O(h)$

casual ↴

1. if $x == \text{Null}$ or $x.\text{key} == k$ then

2. return ~~node~~ x

3. if $k < x.\text{key}$ then

4. return Tree-search($x.\text{left}, k$)

5. else

return Tree-search($x.\text{right}, k$)

$$h = \log n$$

Date _____

ćمیع Iterative Tree search (x, k)

1. while ($x = \text{null}$, $x.\text{key} \neq k$) {
2. if $k < x.\text{key}$
3. $x = x.\text{left}$
4. else
5. $x = x.\text{right}$
6. return x

Tree-Min(x)

while $x.\text{left} \neq \text{null}$ do

$x = x.\text{left}$

3. return x

Tree-Max(x)

1. while $x.\text{right} \neq \text{null}$ do

$x = x.\text{right}$

3. return x

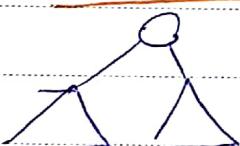
Tree-Min(x)

$\Theta(h)$

1. if $x.\text{left} == \text{null}$

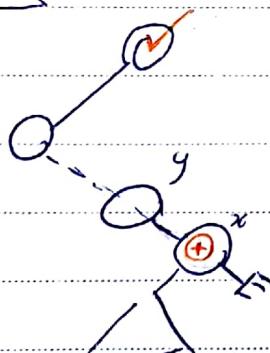
2. return x

3. return Tree-Min($x.\text{left}$)



SUCCESSOR

حُرْهَلَّا فَيْدَى



$x.\text{right} \neq \text{null}$: \rightarrow C1
return Tree-Min($x.\text{right}$)

$x.\text{right} = \text{null}$: \rightarrow C2
 $y = x.P$

while $y \neq \text{null}$ and $x = y.\text{right}$

$x = y$

$y = y.P$

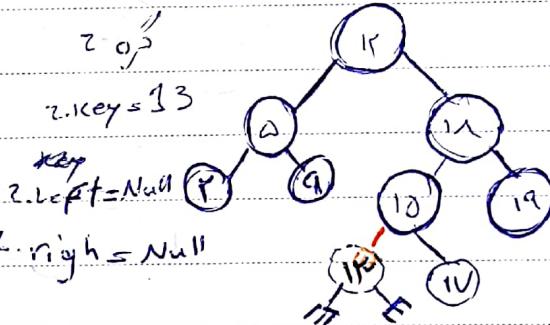
Subject _____
Date _____

Tree-successor(x)

: (Simplification)

- 1 if $x.\text{right} \neq \text{Null}$ then
- 2 return Tree-min($x.\text{right}$)
- 3 else
- 4 $y = x.P$
- 5 while $y \neq \text{Null}$ and $x = y.\text{right}$ do
- 6 $x = y$
- 7 $y = y.P$
- 8 return y

Insert/Delete



Tree-Insert(T, z) $O(h)$

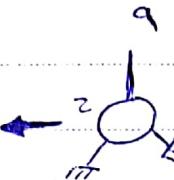
1. $y = \text{Null}$

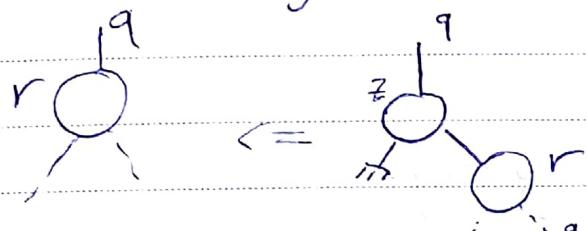
2. $x = T.\text{root}$

3. while ($x \neq \text{Null}$) do
 if ($z.\text{key} < x.\text{key}$) then $y = x$
 $x = x.\text{left}$
 else $x = x.\text{right}$
8. $z.P = y$
9. if ($z.\text{key} < y.\text{key}$)
10. $y.\text{left} = z$
11. $y.\text{right} = z$

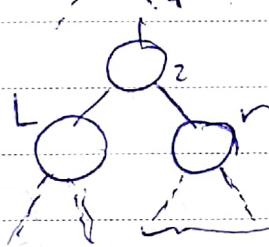
Subject _____
Date _____

Delete خفف

حالت ۱: $q.left = null$ و $q.right = null$  خروجی فرزندی ندارد.



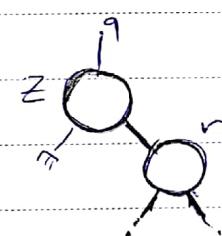
حالت ۲: $q.left \neq null$ و $q.right = null$ خروجی فرزندی دارد.



حالت ۳: $q.left \neq null$ و $q.right \neq null$ خروجی داشته باشد.

عن دردی min
نهایی خلاصه

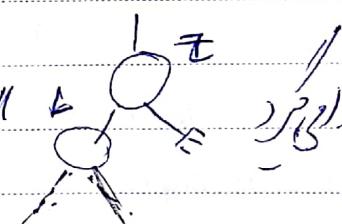
حالت ۱:
 $z.left = null$



نیز خود را باید بحذف کرد.

حالت ۲:

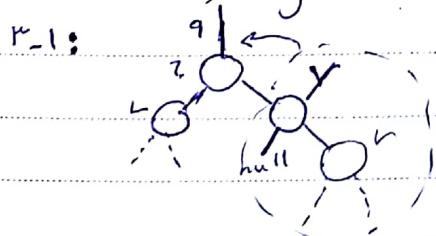
$z.left \neq null$, $z.right = null$



نیز خود را بحذف کرد.

حالت ۳:

$left \neq null$, $right \neq null$



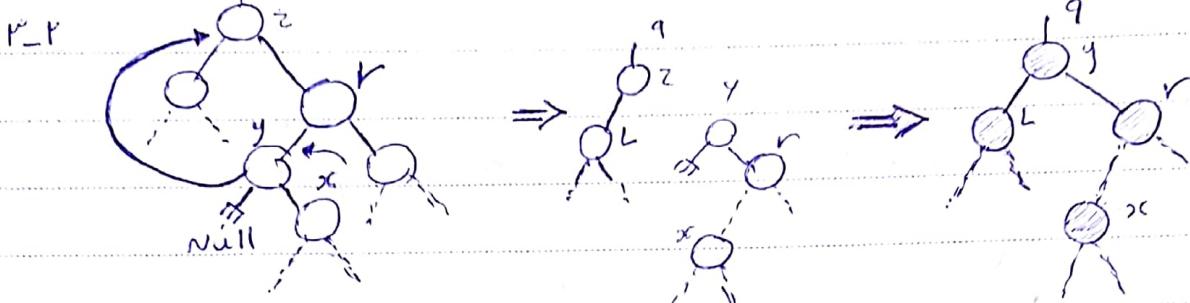
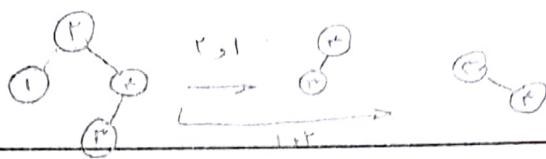
نیز خود را بحذف کرد و قرار بگیرد.

$y.left = L$

(نیز خود را بحذف کرد و خانه خود را بگیرد)

Subject

Date



Transplanting \rightarrow Replacing

Transplant (T, u, v) \Rightarrow $u \rightarrow v$ Delete u by v

1. if $u.p = \text{Null}$ then

2. $T.root = v$

3. else if ($u == u.p.left$)

4. $u.p.left = v$

5. else $u.p.right = v$

6. if $v \neq \text{Null}$ then

7. $v.p = u.p$

Tree - Delete (T, z) $\rightarrow O(n)$

1. if $z.left = \text{Null}$ then

2. transplant (~~Replace~~ $T, z, z.right$)

3. else if $z.right = \text{Null}$ then

4. transplant ($T, z, z.left$)

5. else $y = \text{Tree-Min}(z.right) \rightarrow O(n)$

6. if $y \neq z.right$ then

7. transplant ($T, y, y.right$)

8. $y.right = z.right$

9. $y.right.p = y$

10. transplant (T, z, y)

11. $y.left = z.left$

10

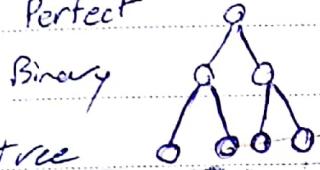
~~$z.z.left.p = y$~~

12. $z.left.p = y$

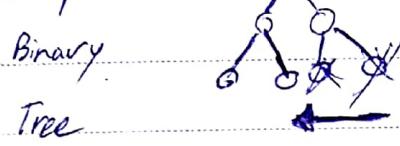
Heap (جعبه)

Complete (Binary) Tree جعبه (باینری)

Perfect



Complete

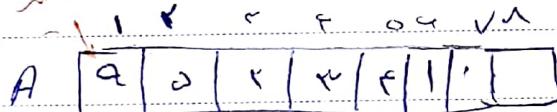
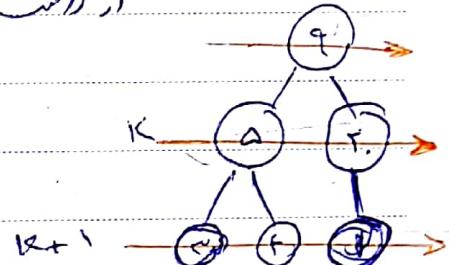


Perfect = درخت باینری
مکمل

بروزی ندارد سر از هر کدامیکی را می‌توان
از راسی برخواهی از آنها را حذف کرد

Heap

max heap = هر کدامیکی بزرگتر از پدر خود باشد



باینری های

i (پدر) parent(i) = $\lfloor \frac{i}{2} \rfloor$
left(i) = $2i$
right(i) = $2i + 1$

heap size [A]

نحوه: length[A]

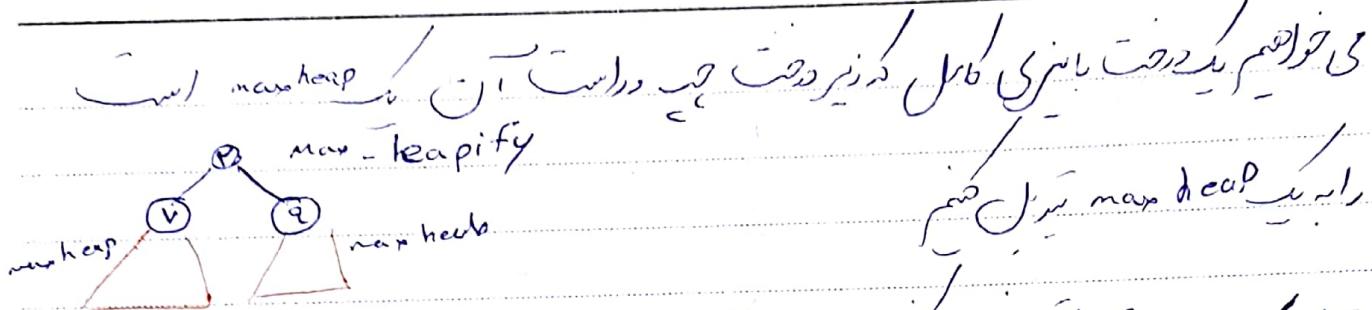
i > heap size [A] \leftarrow i > heap size [A] برای هر کدامیکی از

امروزه هر کدامیکی از اینها را برگردانند

array $\leftarrow \left[\frac{?}{?} \right] \text{ از}$

Subject

Date



$$\log(n) = O(n) \cdot \text{log}(n)$$

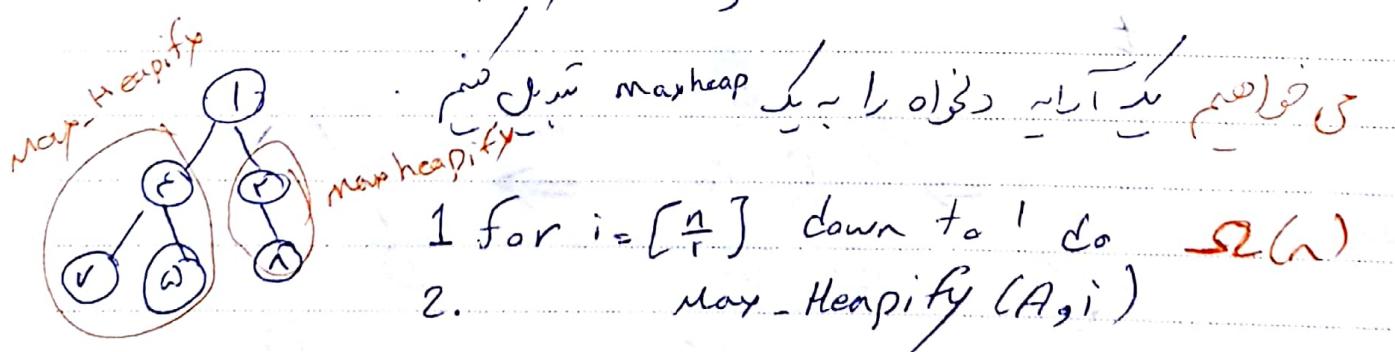


Max_Heapify(A, i)

1. if $\text{left}(i) \leq \text{Heap_size}[A]$ and $A[\text{left}(i)] > A[i]$ then
 2. largest = $\text{left}(i)$
3. else largest = i

4. if $\text{right}(i) \leq \text{heap_size}[A]$, $A[\text{right}(i)] > A[\text{largest}]$ then

5. largest = $\text{right}(i)$
6. if largest $\neq i$ then
7. swap($A[i], A[\text{largest}]$)
8. max_Heapify($A[\text{largest}]$)



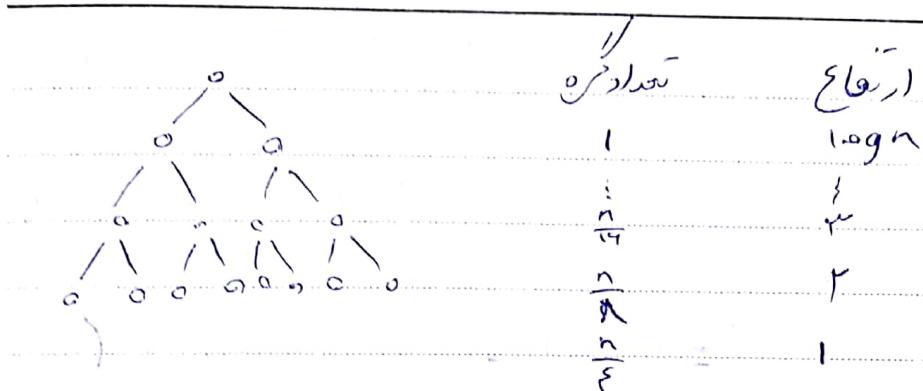
$O(n \cdot \log n)$ tight

$O(n)$ شرطی ایسا \leftarrow tight \leftarrow tight



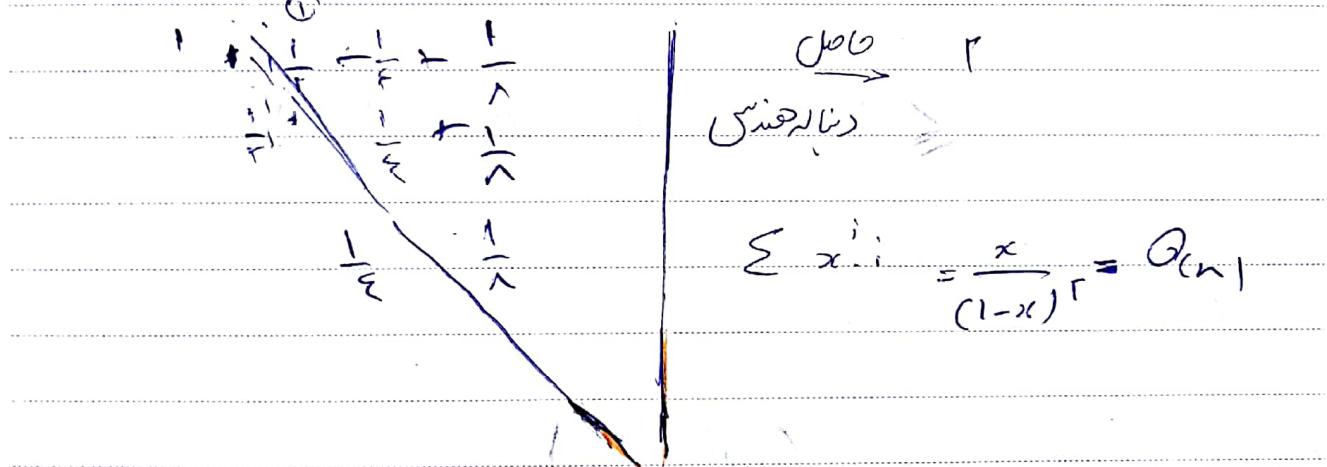
Subject _____

Date _____

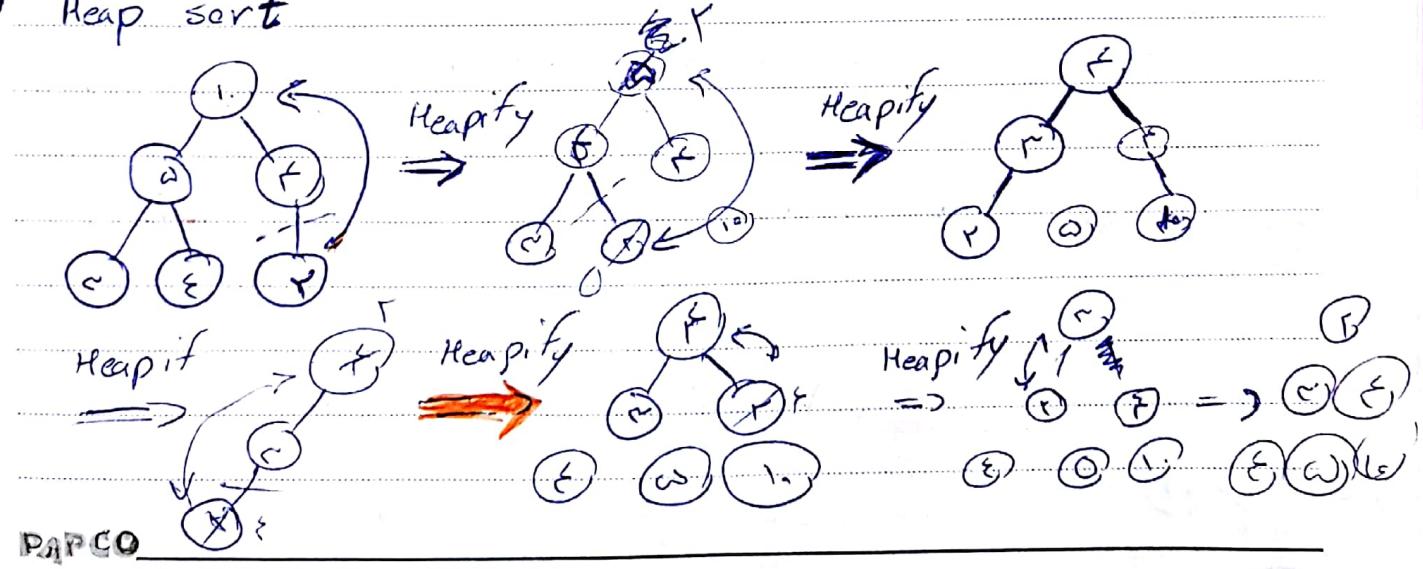


ملاحظة: ~~$C_1 + C_2 + \dots + C_{\log n}$~~

$$C_1 \times \log \frac{n}{2} + C_2 \times \frac{n}{4} + C_3 \times \frac{n}{8} + \dots + C_{\log n} \times \log \frac{n}{2} \approx$$
$$C \sum_{i=1}^{\log n} \frac{n}{2^{i+1}} \times i = \frac{Cn}{2} \sum_{i=1}^{\log n} \left(\frac{1}{2}\right)^i \times i = O(n)$$



Heap sort



PAPCO

Heap-Sort (A)

O(n) 1. Build-Max-Heap(A)

C. n 2. for i = length[A] down to 2

C. n-1 3. swap(A[i], A[n])

C. n-1 4. heap-size[A]--

C. n-1 5. Max-Heapify(A, 1)

$$\text{cost} \leq \sum_{i=1}^{n-1} \log i \leq \log[(n-1)!]$$

$$\log n! = \Theta(n \log n) \longrightarrow \star$$

Balanced Binary Search Tree

red-black tree

هر گرهی دویتی دارد و هر گرهی دویتی دارد و هر گرهی دویتی دارد و هر گرهی دویتی دارد

: Red-black tree

① هر گرهی دویتی دارد

② ساده ترین صورت (Null) گرهی دارد

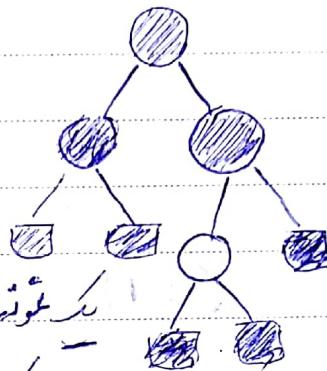
③ اگر گرهی دویتی دارد هر دویتی دارای فرزندان می باشد

④ برای هر گرهی دویتی دارای سه صورتی دارد: یکی بزرگتر از همه دیگرها، دیگری کمتر از همه دیگرها و دیگری برابر با همه دیگرها

Subject _____

Date _____

Null = بیک
Null جو = بیک جو



جواب = خاسو

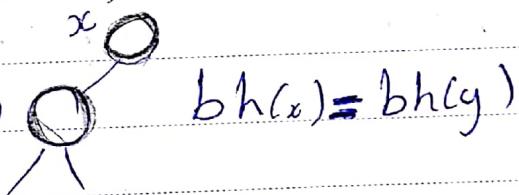
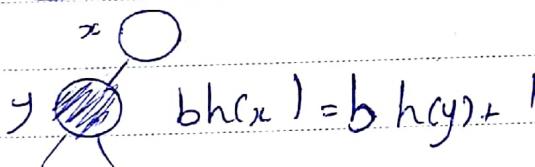
Red-Black ایجاد کردن

کل سریع ترین برای طول بسته دستگاه

black height = bh(x)

$$h(x) \leq bh(x)$$

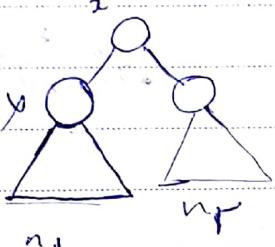
(جواب عرضی)



$$\rightarrow bh(y) \geq bh(x) - 1$$

log₂(n+1) ارتفاع جو تاں (1/b) Red-Black

Subject _____
Date _____



الثوابت المطلوب إثباتها: $n \geq r^{bh(x)} - 1 \geq r^{bh(y)} - 1 \geq r^{bh(z)} - 1$

$$n \geq r^{bh(y)} - 1 \geq r^{bh(x)} - 1$$

$$n = n_l + n_r + 1 \geq (r^{bh(x)} - 1) + (r^{bh(y)} - 1) + 1$$

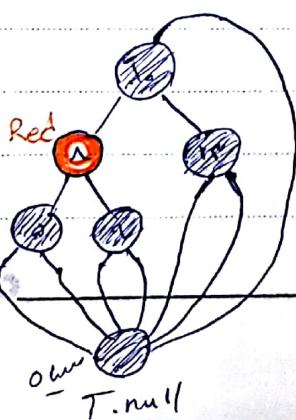
~~$$n \geq r^{bh(x)} - 1$$~~

$$h=0 \rightarrow n=1 \text{ (جاء من المقدمة)}$$

$$\therefore n \geq r^{bh(T)} - 1 \geq r^{h(T)/k} - 1$$

$$\Rightarrow h(T) \leq k \log(n+1)$$

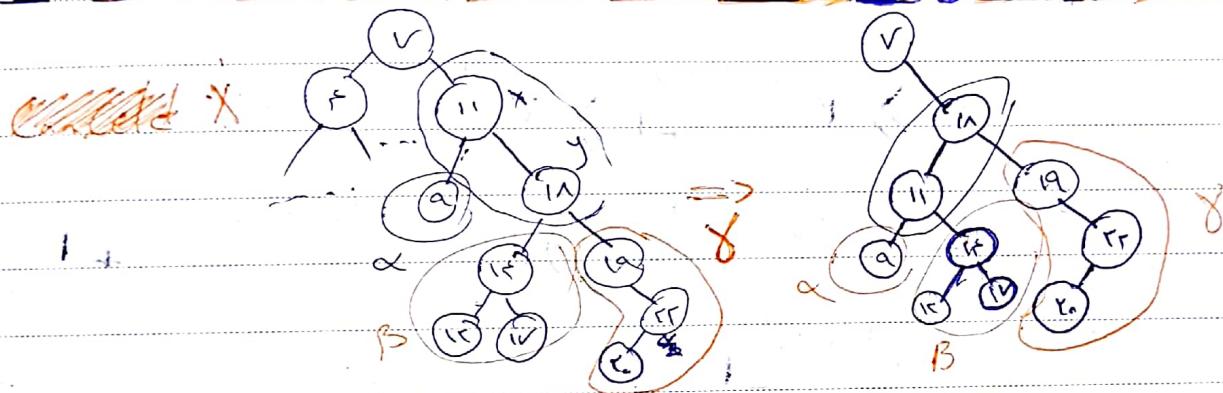
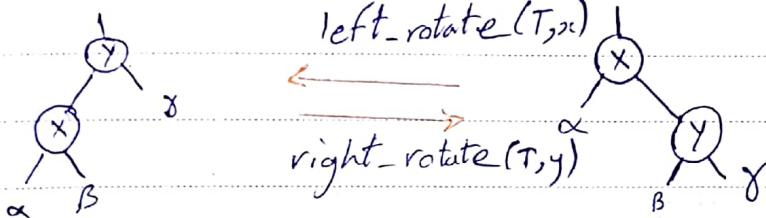
• AVL tree - search (جاء من Red-black Tree)



الرجوع إلى
الخلف

بعد رجوع إلى عنصر ديناميكي يغير
بياناته في العقدة التي
تم إدخالها في العقدة

Rotation



$\text{left_Rotate}(T, x)$ $O(1)$

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 $x == x.p.\text{left}$

9. $x.p.\text{left} = y$

10. else $x.p.\text{right} = y$

11. $x.p = x.y$

12. $y.\text{left} = x$

RB-Insert(T, z)

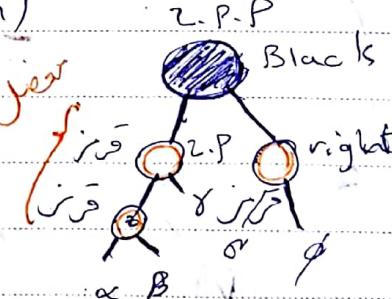
initial red-black b. node by

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

15. RB-Insert-Fixup(T, z)

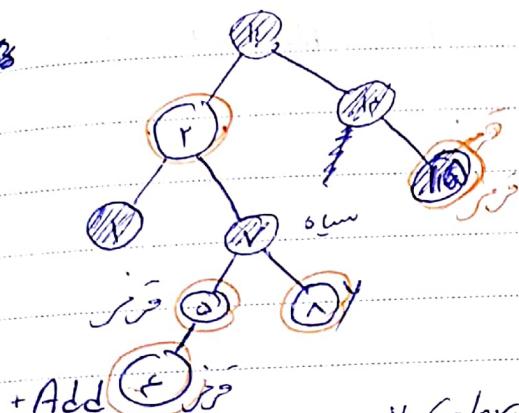
→ RB-Insert-Fixup(T, z) $O(\lg n)$

1. while $z.p.\text{color} == \text{Red}$ do
2. if $z.p == z.p.p.\text{left}$ then
3. $y = z.p.p.\text{right}$
4. if $y.\text{color} == \text{Red}$ then



ویرجی دست رفته سلسلت.

RB



: Case 1

حیزاست y , $y = y.P.P.right$

برای حل مشکل بزرگ شدن درخت باید

+ Add \leftarrow

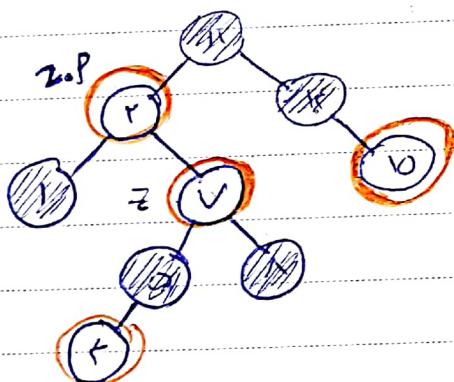
$y.color = Black$

$= Z.P, Z$

$Z.p.color = Black$

$Z.p.p.color = RED$

$\underline{Z.P} \quad Z = Z.P.P$



: Case 2

حیز است y , $y = Z.P.P.right$

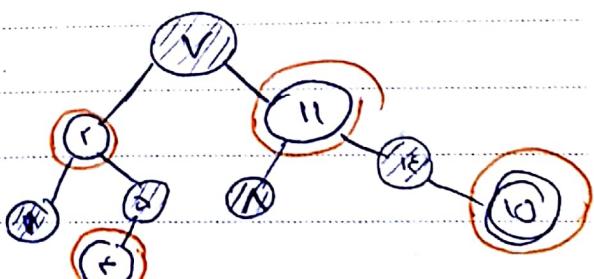
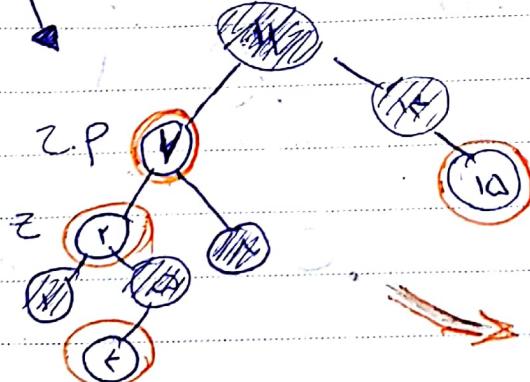
برای حل مشکل بزرگ شدن درخت باید $Z.P$ را



: Case 3

برای حل مشکل بزرگ شدن درخت y , $y = Z.P.P.right$

$Z = Z.P$: Case 3 ~ Case 2
left_rotate (T, Z)



Subject _____

Date _____

z.p.p.color = Red

: Case 3 ، فی الحالات اینجا

z.p = Black

right-rotate (T , z.p.p)

(لارجیل پر) راست روشن (Right) Rotate

RB-Insert-Fixup (T , z)

1. while z.p.color == Red do {
2. if z.p == z.p.p.left then
3. y = z.p.p.right
4. if y.color == Red then // Case 1
5. y.color = Black
6. z.p.color = Red
7. z.p.p.color = Red
8. z = z.p.p
9. else if z == z.p.right then // Case 2
10. z = z.p
11. left-Rotate (T , z)
12. z.p.color = Black // Case 3
13. z.p.p.color = Red
14. Right-Rotate (T , z.p.p)

15 else (same as then ---) → right-left, گزینه ۱،

گزینه ۲،

16 $T_{\text{Root}}.color = \text{Black}$

PAPCO _____

PA

Subject _____
Date _____

loop invariant:

یک گزاره که در طول چند جریان یعنی دست است.

بررسی محتوی علی دفعه Red-Black

loop invariant

ا) جزوی از قرز است

ب) اخر دست آنها Z.P زیر دست است

c) اخر دست ویرجی Red-black (تفصیل) فقط این دست ویرجی را تفهی فرموده است

امروزی در (تفهی) نهاد لیش این است در جریان است و قرز است. اخر دست ویرجی علی (تفهی)

نه لیش این است در Z.P اخر در قرز هست.

حالت اولیه (اگر اول)

بینهایت کمتر و بینهایت بزرگ داشتی اولین گزاره خارجی while دست است.

عمل از فرایانی P ، Fixup ، Color-Red

a) بینی است در دست است b) اخر P.Z ریس برآورد چون Z.P زمانی نیز برآورده از اول

رسیز بوده رسیده است اخر P.Z رسیدن بینهایت گزاره ب

با شناخت هست است

Subject

Date

خواهد
خواهد بـ انتهايی از سرهای دوست (اضافه شود و بعد آن فرز است و سار خواهد

و ساختار دوست می خواهد. بنابران در عبارت های ساده میتوان ساده است (دریجی دفعه

می شود). دریجی دفعه عربی است دفعه نیشود و چون فقط دفعه دفعه نیشود و فرز است

نمایی دفعه است دریجی دفعه نیشود این است دفعه رسمی است.

نهایی دفعه است دریجی دفعه نیشود این است دفعه رسمی است.

(فرز دفعه) است دفعه رسمی است (و در صورت دفعه فرز نیشود) دفعه

و از دفعه نیشود با P فرز نیشود دفعه نیشود، بالعكس

Termination (کلیه) (تیر)

دریجی دفعه نیشود اگر دریجی دفعه نیشود (لیکن) این است دفعه رسمی

است و فرز است. در این صورت دفعه T به از طبق while

عمل را حل دعماً دریجی های برقرار است.

اگر دریجی دفعه نیشود (لیکن) این است دفعه و قریب است. دفعه حلقه ای while

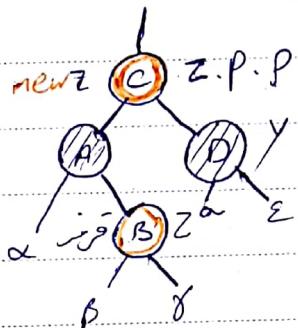
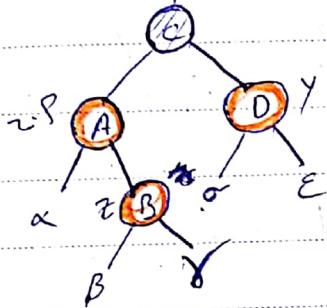
تعجب نیشود P است سه دفعه خواهد دفعه نیشود.

maintenance

John C. W.

امراض ایمنی while مرض ایمنی (C) (B) (a)

Case 1:



جزء اسـتـرـیـلـیـزـیـشنـ (S.P.P) دوـنـدـرـلـرـ (D.P.D)

جـ ٢.٣.١ جـ ٢.٣.٢ جـ ٢.٣.٣ جـ ٢.٣.٤ جـ ٢.٣.٥

أحرز جيداً سبعة أدنى (D. P. 6) سنتين وجزءاً من سنتين

حرازه b درست است و در میر پ. 2 غیر قانونی است باید حداکثر حرازه b درست است

جزایر ②: عدهایی از مرئی های ۱۶۲ مقطع) می شود - بجز این بسیار

نیز میتواند (T null) بنابراین از طبق خارجی سوم و پنجم درست نباشد.

اگر ویرجی (۱) برقرار نہیں ہے:

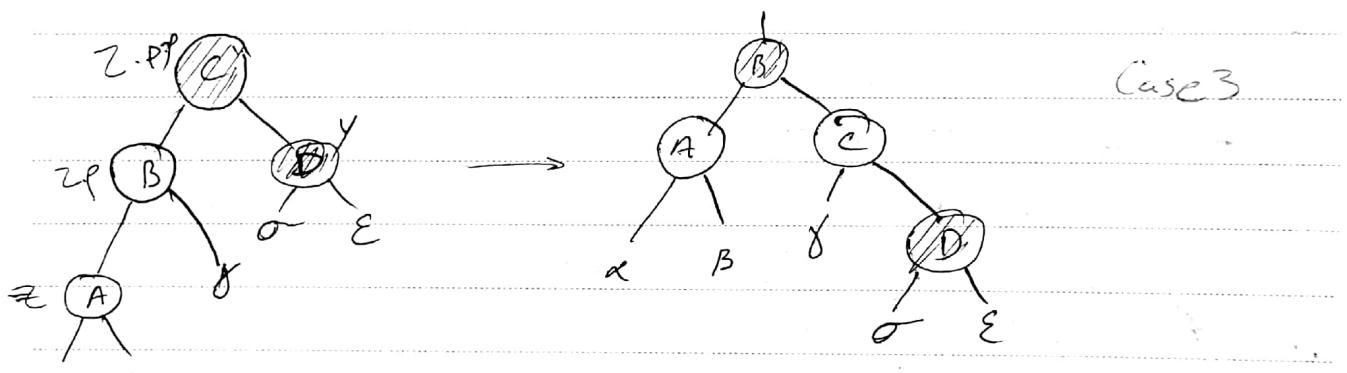
بچھے بھرائی حرد Case 1 دوستی کی صورت میں داد دینے والے افراد

ویرجی @ برقرار نہیں ہم برقرار رہتے۔

ویرجی ۱، ۲، ۳ صورت بھی برقرار رہتے۔

اگر ویرجی (۱) فتح علی (پورا) ہے تو ز = Z.P.P

کوئی نہیں دیکھی ویرجی (۴) میں اس تھیں سر (اصحی) کے ز پ کے جدید ختنے میں



while Z.p.color == Red do

{

if قرمز

Case 1

else if فربنیہ

Case 2

Case 3

Subject _____

Date _____

R.B.-Transplant(T, u, v)

Red-Black \rightarrow (irr.)

1. if $u.p = T.\text{null}$ then
2. $T.\text{root} = v$
3. else if $u = u.p.\text{left}$ then
 4. $u.p.\text{left} = v$
5. else $u.p.\text{right} = v$
6. ~~else~~
 $v.p = u.p$

R.B.-Delete(T, z)

1. $y = z$
2. $y.\text{original_color} = y.\text{color}$
3. if $z.\text{left} == T.\text{null}$ then
4. $x = z.\text{right}$
5. R.B.-Transplant($T, z, z.\text{right}$)
6. elseif $z.\text{right} == T.\text{null}$ then
7. $x = z.\text{left}$
8. R.B.-Transplant($T, z, z.\text{left}$)
9. else $y = \text{Tree-Min}(z.\text{right})$
10. $y.\text{original_color} = y.\text{color}$
11. $x = y.\text{right}$
12. if $y.p == z$ then
13. $y.p = z$
14. else // R.B.-Transplant($T, y, y.\text{right}$)
15. $y.\text{right} = z.\text{right}$
16. $y.\text{right}.p = y$

Subject

Date

17.

RB - Transplant (T, z, y)

18

$y.left = z.left$

19

$y.left.p = y$

20

$y.color = z.color$

21 if $y.original_color == \text{Black}$ then

22 RB - Delete - Fixup (T, x)

١) y سیاه فرزند x را دارد

٢) y سیاه فرزند x ندارد

٣) x سیاه است

٤) x سیاه نیست

٥) x سیاه است و x که x را پدر می‌داند سیاه است

Black - Black \rightarrow (double black)

x is
Red - Black

RB - Delete - Fixup (T, x)

1. while $x \neq T.root$ and $x.color = \text{Black}$ do

2. if $x == x.p.left$ then

3. $w = x.p.right$

4. if $w.color = \text{Red}$ then

5. $w.color = \text{Black}$

ART

Subject

Date

6.

$x.p.color = \text{Red}$

7.

$\text{left_Rotate}(T, x.p)$

8.

$w = x.p.right$

Case 1

Case 2

Case 3

Case 4

9. if $w.left.color == \text{Black}$ and $w.right.color == \text{Black}$ then

10.

$w.color = \text{Red}$

11.

$x = x.p$

Case 2

SC

to Case 1

12. else if $w.right.color = \text{Black}$ then

13.

$w.left.color = \text{Black}$

14.

$w.color = \text{Red}$

Case 3 → Case 4

15.

$\text{Right_Rotate}(T, w)$

16.

$w = x.p.right$

IV.

$w.color = x.p.color$

17.

$x.p.color = \text{Black}$

18.

$w.right.color = \text{Black}$

19.

$\text{left_Rotate}(T, x.p)$

20.

$x = T.root$

21.

$x = T.root$

22.

$x = T.root$

23.

$x = T.root$

24.

$x = T.root$

25.

$x = T.root$

26.

$x = T.root$

27.

$x = T.root$

28.

$x = T.root$

29.

$x = T.root$

30.

$x = T.root$

31.

$x = T.root$

32.

$x = T.root$

33.

$x = T.root$

34.

$x = T.root$

35.

$x = T.root$

36.

$x = T.root$

37.

$x = T.root$

38.

$x = T.root$

39.

$x = T.root$

40.

$x = T.root$

41.

$x = T.root$

42.

$x = T.root$

43.

$x = T.root$

44.

$x = T.root$

45.

$x = T.root$

Case 1

$w = \text{Red}$

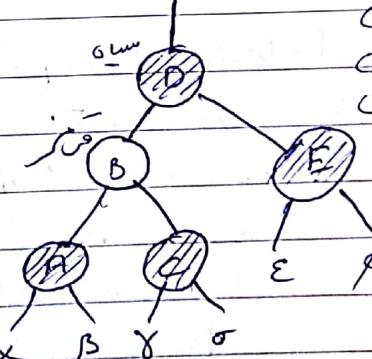
$x = \text{Black}$

$w = \text{Black}$

$x = \text{Red}$

$w = \text{Black}$

⇒

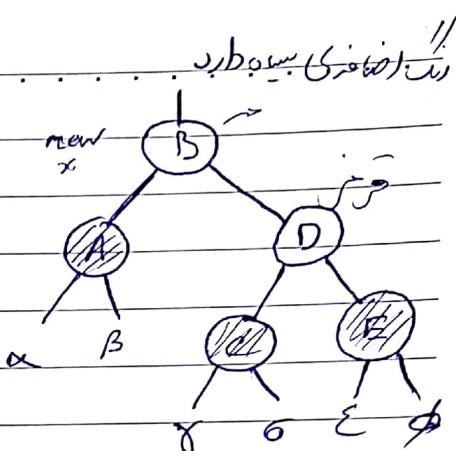
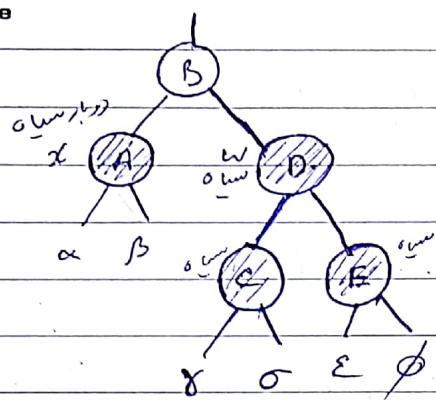


ART

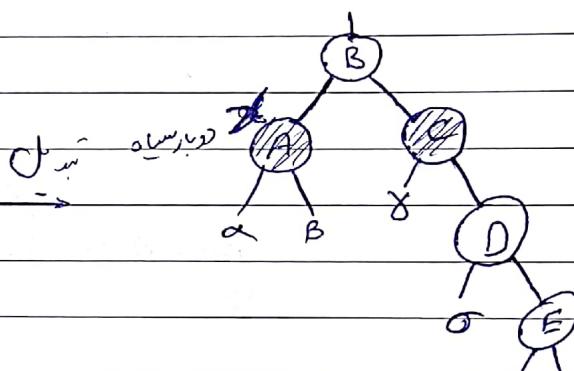
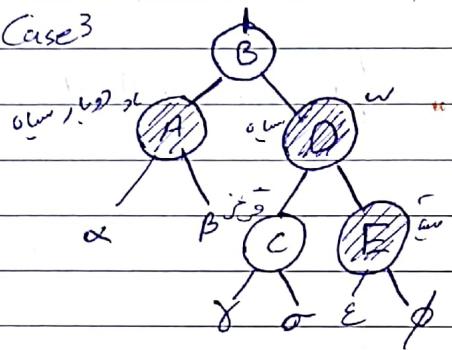
Subject

Case 2

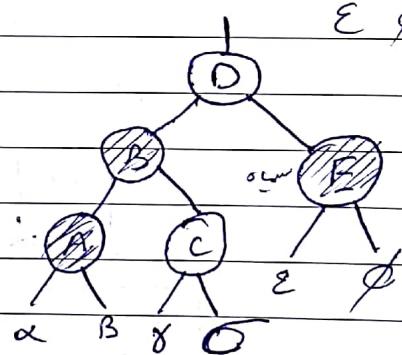
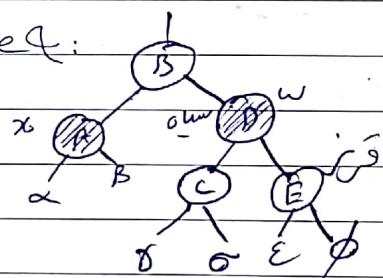
Date



Case 3



Case 4:



Augmenting Data structures

ایجاد اطلاعات بین تابعی داده های سریع برای کاربران و افزایش را ان جایز می کند

Red Black Tree

جیزا

ART

۴۲

Subject

Date

then (lower) μ selection

then μ_{avg} with μ_{avg} : ranking.

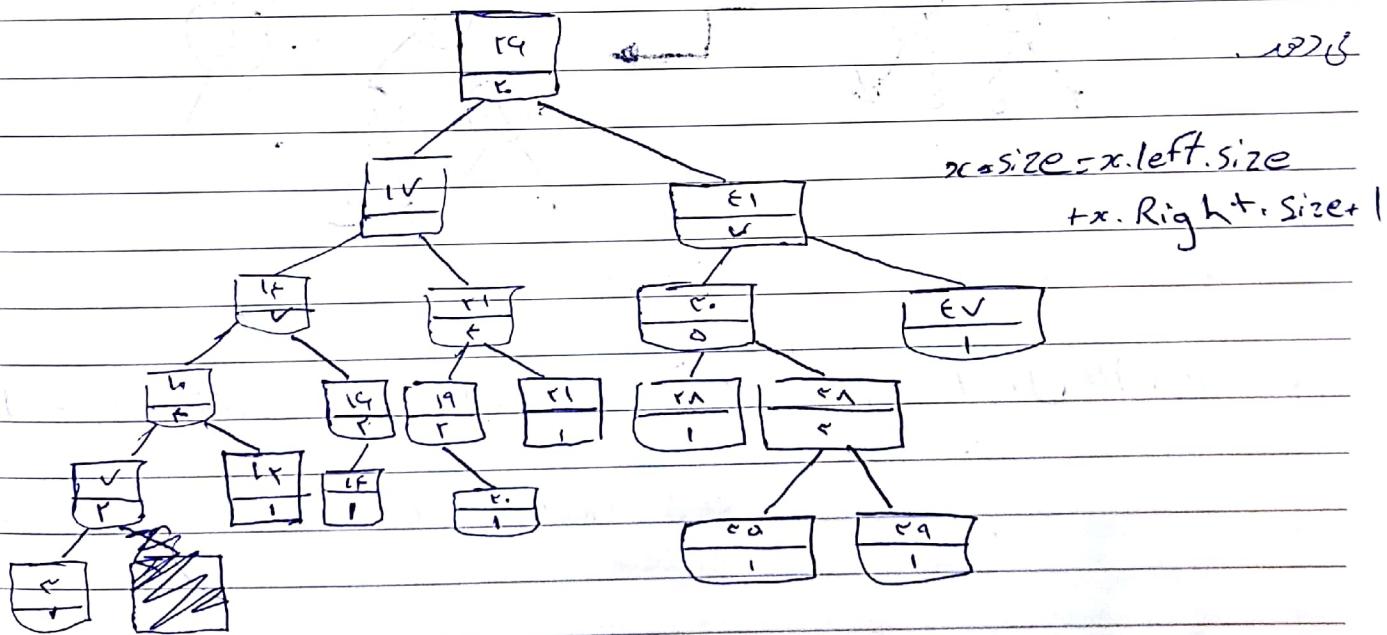
بامداده (نیز) سری دستان (O(nlogn)) حل یافته

مودرلیو^ن (O_n) حمل جردن

Red-Black n- selection using J

$\frac{m}{k} \omega_0$, $m=1V$, $k=1V$

جزئیاتی مورخی جزئیاتی size (اضافی) نیست و تعداد جزو های زیرحشته برسیده اسان



T.null.size = 9

Subject

Date

OS-select(x, k) $O(\log n)$

1. $r = x.\text{left.size} + 1$

2. if $k = r$ then

3. return x

4. else if $k < r$ then

5. return OS-select($x.\text{left}, k$)

6. else return OS-select($x.\text{right}, k - r$)

$O(n \log n)$ RB-tree insert

: Red-Black tree Rankings: $\Omega(n^2)$

OS-Ranking(T, x) $O(\log n)$

1. $r = x.\text{left.size} + 1$

2. $y = x$

3. while $y \neq T.\text{root}$ do

4. if $y == y.p.\text{right}$ then

5. $r = r + y.p.\text{left.size} + 1$

6. ~~return r~~

6. $y = y.p$

7. return r

Size Selection

: 2nd fix

: Fixup \Rightarrow Fixup : 2nd fix

ART

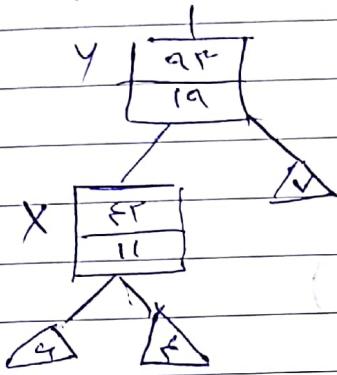
(e)

Subject

Date

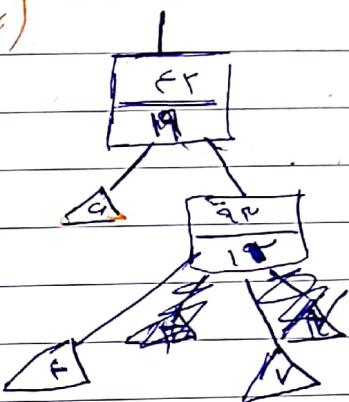
١) ایجاد: از جمله حجت های جبرین است - $O(n^2)$ size

Rotate \vec{v}_1 around \vec{v}_2 to get \vec{v}'_1 such that $\vec{v}'_1 \perp \vec{v}_1$: fix up (1)



left-Rotate(T, x)

right-rotate(T, y)



left-Rotate(T, x)

~~مسار~~ مسار $\{ y.size = x.size \}$

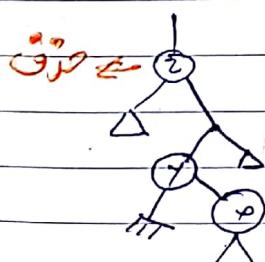
$$\text{left-rotate } x \quad | \quad x.\text{size} = x.\text{left.size} + x.\text{right.size} + 1$$

فیکار علی خنفی

۱) ای حصہ: زحل اور سیاروں کا حصہ (زہر حلقہ سے) ترکیب از

$\mathcal{O}(\log n)$

Open Source Fixup



Subject: _____
Year. _____ Month. _____ Date. _____

Interval Tree

o ii Interval $[t_l, t_r]$

$$\{x \in R : t_l \leq x \leq t_r\}$$

o ii $i.\text{low} = t_l, i.\text{high} = t_r$

init ≠ φ \rightarrow $i.\text{int}.$ (جواب) overlap $i.i$ و $i.i$ دویت

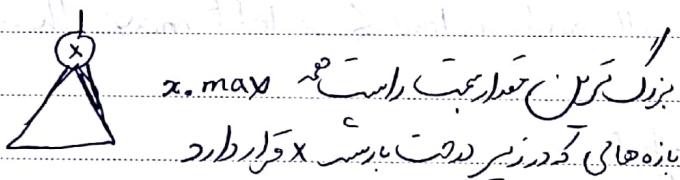
لور

~~لور~~ $i.i$ دویت

$(i.\text{high} < i.\text{low})$ \rightarrow $i.i$ دویت

$(i.\text{low} > i.\text{high})$ \rightarrow $i.i$ دویت

x \rightarrow $x.\text{int}$. Red-Black Tree (لر بکس) \rightarrow $x.\text{max}$



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

$$x.\text{key} = x.\text{int}.x.\text{low}$$

Subject: _____
Year: _____ Month: _____ Date: _____

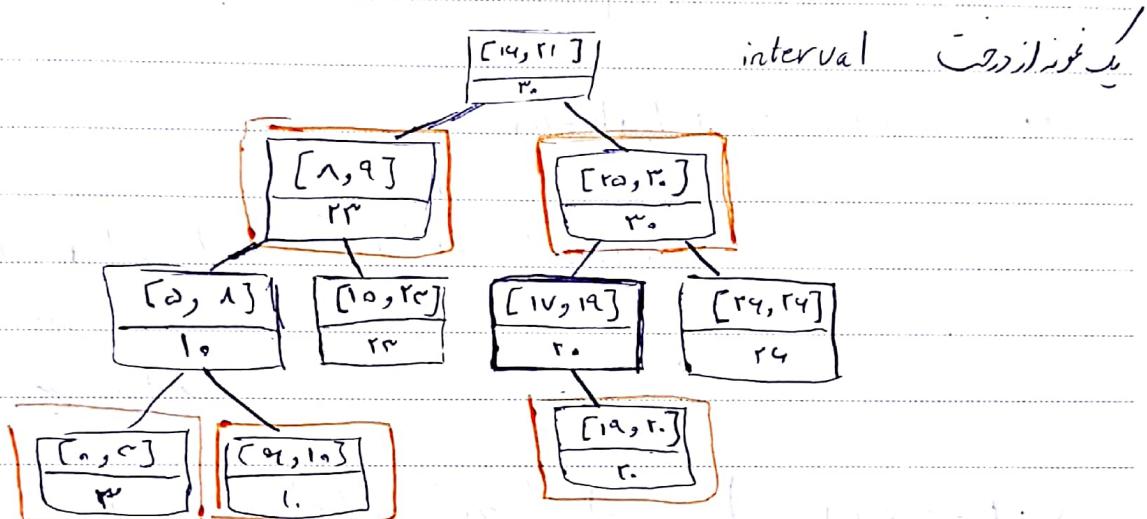
Interval-Insert(T, x)

: ایجاد اینترval

Interval-Delete(T, x)

Interval-search(T, i)

..... ایجاد اینترval برای جستجوی داده ها



Interval-search(T, i)

$O(\log n)$

1. $x = T.\text{root}$
2. while $x \neq T.\text{null}$ and x^{int} does not overlap interval i do
3. if $x.\text{left} \neq T.\text{null}$ and $i.\text{low} \leq x.\text{left}.\text{max}$ then
4. $x = x.\text{left}$
5. else $x = x.\text{right}$
6. return x

..... ایجاد اینترval برای جستجوی داده ها

Subject _____
Date _____

لیست داشتی:

نایت حلقة دیگر درست تا زمانی که در طرف دار را نمایش نمینه
while دستور طرف دار
که در زیر دست باریست و نزدیک درست دارد.

عمل لازم اخراجی حلقاتی Initialization while (حالت پایه)

برصرت بعلی درست است جمله ای که نیز مکمل درست تا است
x = T.root +
حالت تغییری: maintenance

آخر رسانیدی بیکار از حلقاتی while نایت حلقات درست باشد رسانیدی بکار بردی شوند

نایت حلقات درست است.

حالت داریم: ۱) اگر صفت x.left == null ۲) اگر صفت else درست است

در در صورت باندهای طاب در زیر دست left میست دامن max

چنانچه و جو در دامن باشد در زیر دست راست است این نایت حلقات اتفاق

بازهای دستور دست جمی باشند high

i. low < x.left.max = i' high : ۲) در حالی که if برقرار باشد

Subject

Date

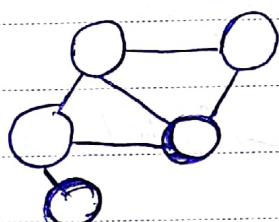
اف) $i \leftarrow i \cdot low < i \cdot high$ باره از دیگر میتوانیم بداند.

ب) $i \leftarrow i \cdot high < i \cdot low$ باره از دیگر میتوانیم بداند.
لکه هر یکبار زیر رخت راست

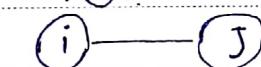
$G = (V, E)$
ما له چونه سال ها

الترمیم جای حرف

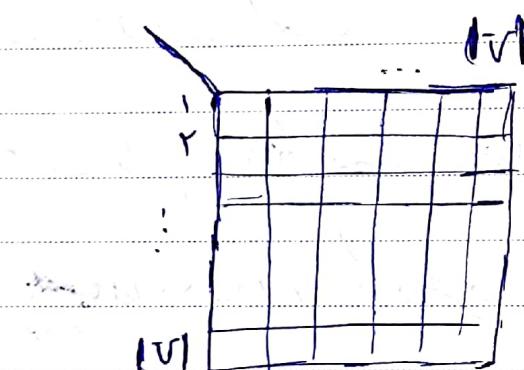
$$E \subseteq V \times V$$



سونجات



ست در



$$\text{مانوس} \text{ متسابق} = \Theta(|V|^2)$$

مانوس متسابق

$$A[i, j] = \begin{cases} \infty & \text{اگر زیر یک مسیر ندارد} \\ 1 & \text{اگر زیر یک راه دارد} \end{cases}$$

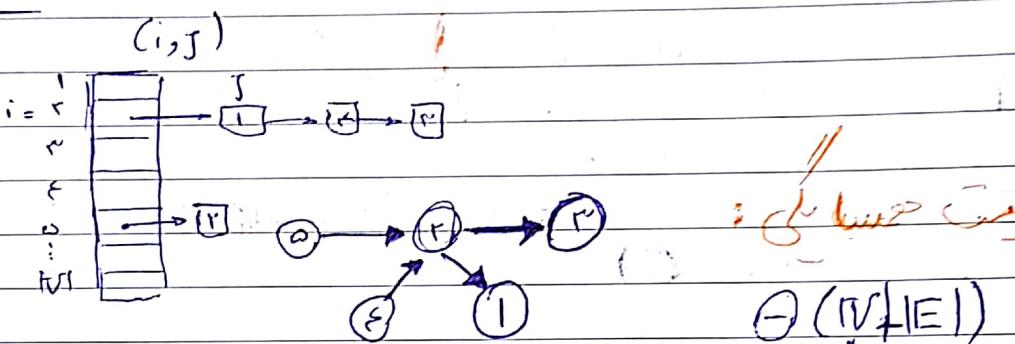
$$A[i, j] = \begin{cases} \infty & \text{اگر زیر یک مسیر ندارد} \\ w(i, j) & \text{وزن یک مسیر (j وا) } \end{cases}$$

Subject

Date

IVI، IV-II

حترالن تعدادیل حملین بت در =



بغض فاعل بینبورون

$$IVI - 1 \leq |E| \leq |V| \times (|V| - 1)$$

↓
تماریل

تماریل حملی خامس اس sparse راو

خامس راو

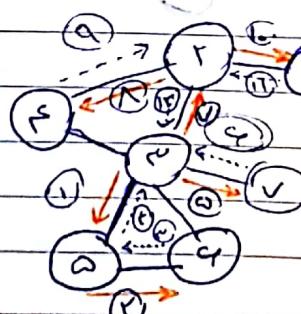
جست و جوی اولیه (Death First search) DFS

جست و جوی سهوارل

) BFS

تو مر جر ای دھنسی بر و بہ کتابی ای دھنسی
اکھر پیا نمودی بر جر دیسی دھن

ART



۵۱