

"ساختہ ان دادہ و الگوریتم"

تابع:

Introduction to Algorithms C.L.R.S.

۱۳۸۸ - دادہ ساختہ ریکارڈر الگوریتم - دکتر قدیر - استاد رات مالی -

الگوریتم؟ راہی برائی حل مسئلہ سے کے لوب مای سینا ہے نسخہ جنم شہ مزدھوں سے۔

ساختہ ان دادہ؟ ذریع نگاری و ذصیرہ ای اطلاعات درودی و دادہ ما (datastructure)

تکمیل زبانی الگوریتم ہاں زجن منظور تعداد دستور ہائی استاد را طہ الگوریتم فرستادہ ہی ٹوں۔

* بیان الگوریتم جائیں کہ رسود کلام انجام گئیں۔

زمان = تعداد دستورات پایہ

زمان = تابع از اندازہ درودی

جزئیہ سندوار

1. c_1 1. $x \leftarrow 0$

$n+2$ c_2 2. for $i \leftarrow 0$ to n do

$n+1$ c_3 3. $x \leftarrow x + 2$

$$T(n) = 1 \times c_1 + (n+2) \times c_2 + (n+1) \times c_3 = a \cdot n + b$$

نیاں الگوریتم
کے تابع درجہ ۱ بحسب n اکبر دردی ۲ نام بیشتر ہے جنہیں ہم ہماری زبان میں بارہی نہیں۔

(linear)

۹۷، ۴، ۲۶

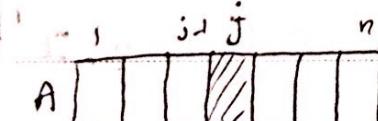
صلبہ دوم

: (Insertion sort) تکمیل الگوریتم مرتب سازی در جریب

و دردی: $A \approx \sqrt{n}$ در حقیقی درجہ

مرتب جسے $A[i]$ کے قیمتیں $A[1], A[2], \dots, A[n]$ کے صورت مرتباً شروعی میں موردی

$$\forall 1 \leq i < n \quad A[i] < A[i+1]$$



مرتب شدہ صورتی

sort

۱. اول فرضی کنیم که عدد ارایه شده درستی لا اینها درست
 ۲. عدد ارایه شده ایجاد مرتب شده اما نهایی نگیرد و باز پرسی کنیم.
 ۳. عدد ارایه شده ایجاد مرتب شده اما نهایی نگیرد و باز پرسی کنیم.

Insertion-Sort (A, n)

1. for $j \leftarrow 2$ to n do
2. key $\leftarrow A[j]$
3. $i \leftarrow j - 1$
4. while $key < A[i]$ and $i > 0$ do
 - $c_{\text{swap}} \sum_{j=2}^i t_j + 1$
5. $A[i+1] \leftarrow A[i]$
6. $i \leftarrow i - 1$
7. $A[i+1] \leftarrow key$

جزئیات مورد نظر

c_{swap} n

c_v n-1

c_p n-1

c_{swap} $\sum_{j=2}^i t_j + 1$

c_{swap} $\sum_{j=2}^i t_j$

c_v $\sum_{j=2}^i t_j$

c_p n-1

for $j = 1$ to n , while $t_j > t_{j+1}$ do

$$T(n) = c_{\text{int}} + c_v(n-1) + c_p(n-1) + c_{\text{swap}} \sum_{j=2}^n (t_j + 1) + c_a \sum_{j=2}^n t_j$$

$$c_v \sum_{j=2}^n t_j + c_v(n-1)$$

ازمی مکانیکی برای $t_j < t_{j+1}$:

best case

جنسینی حالت ($t_j = 0$) $T(n) = c_{\text{int}} + b$ درست

وقت رفیعی (کمترین ترتیب) در دری از همان اینجا امرت شده صورتی باشد.

worst case

جنسینی حالت ($t_j = j-1$) $T(n) = c_{\text{int}} + c_v(n-1) + c_p(n-1) + c_{\text{swap}}(n-1)/2 + c_a n(n-1)/r + c_a n(n-1)/r + c_v(n-1) = c_{\text{int}} + c_v(n-1) + c_p(n-1) + c_{\text{swap}}(n-1)/2 + c_a n(n-1)/r + c_a n(n-1)/r + c_v(n-1)$

$$\sum_{j=2}^n t_j = \sum_{j=2}^n j-1 = \frac{n(n-1)}{2}$$

وقت کمترین ترتیب از همان اینجا امرت شده بزرگی نباشد رفیعی داشت.

کامپیو

Subject: _____
Year _____ Month _____ Day _____

average case

$$\bullet \text{حالات متوسط} \quad t_j = \frac{j-1}{n}$$

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

ویرایش

(Bubble sort) مرتب سازی حبابی

Bubble-sort(A, n)

کسرار هزینه

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

$c_1 \quad n$

2. for $j \leftarrow 1$ to i do

$c_r \quad \sum_{i=1}^{n-1} (i+1)$

3. if $A[j] > A[j+1]$ then

$c_r \quad \sum_{i=1}^{n-1} i$

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

$i \leq 1$

9V, 7, 31

جواب

$$\begin{aligned} \text{bubble sort } T(n) &\leq c_1 \cdot n + c_r \sum_{i=1}^{n-1} (i+1) + c_r \sum_{i=1}^{n-1} i = c_1 \cdot n + c_r \frac{(n+1)(n+1)}{2} + \\ &c_r \frac{n(n-1)}{2} = a \cdot n^2 + b \cdot n + c \end{aligned}$$

$$\begin{aligned} \text{الگوریتم} &= a \cdot n^2 + b \cdot n + c' \\ \text{رسانید} & \quad \text{درست رسانید} \end{aligned}$$

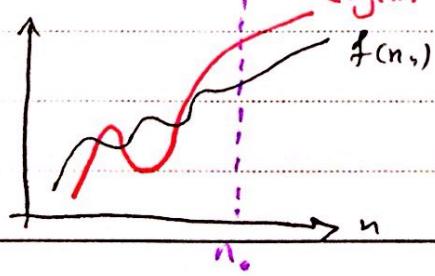
رشد توانع: حداقل شرط

w, o, θ, Ω, O

little omega, little o, theta, big omega, big o

big O

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



رسانید $f(n)$ بعد از n_0 حد بالا $g(n)$ را دارد

- $\exists f(n) \geq c \cdot g(n)$ برتری مساوی $f(n) \geq c \cdot g(n)$

۱۲

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

$$\lim_{n \rightarrow \infty} \frac{\sqrt[n]{f(n)}}{n}$$

اگر $C \neq 0$ ، اگر $\lim_{n \rightarrow \infty} \frac{\sqrt[n]{f(n)}}{n} = C$

$$\lim_{n \rightarrow \infty} \frac{\sqrt[n]{f(n)}}{n} = C$$

$$n \in O(\sqrt[n]{f(n)}) \Rightarrow \dots$$

$$n \in O(\sqrt[n]{f(n)}) \Rightarrow \begin{cases} n \in O(\sqrt[n]{f(n)}) \\ n \notin O(\sqrt[n]{f(n)}) \end{cases}$$

$$\log n \in O(n)$$

برای دلیل: $n \in O(\sqrt[n]{f(n)}) \Leftrightarrow \exists C > 0, \exists n_0 > 0 | \forall n \geq n_0, n \leq C \cdot (\sqrt[n]{f(n)})$

$$n \leq C \cdot \sqrt[n]{f(n)}$$

$$\frac{1}{n} \leq \frac{1}{C}$$

لذا $\log n \leq C$

$$O(\sqrt{n}) \Rightarrow n \in O(\sqrt{n}) \quad n \leq C \cdot \sqrt{n} \quad \sqrt{n} \leq C$$

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

$f(n) \geq Cg(n)$ میگویند $f(n) \geq Cg(n)$ میگویند

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

$$\lim_{n \rightarrow \infty} \frac{\sqrt[n]{f(n)}}{n}$$

اگر $C \neq 0$

$$\lim_{n \rightarrow \infty} \frac{\sqrt[n]{f(n)}}{n} = C$$

$$n \in \Omega(\sqrt[n]{f(n)})$$

$$n \in \Omega(\sqrt[n]{f(n)})$$

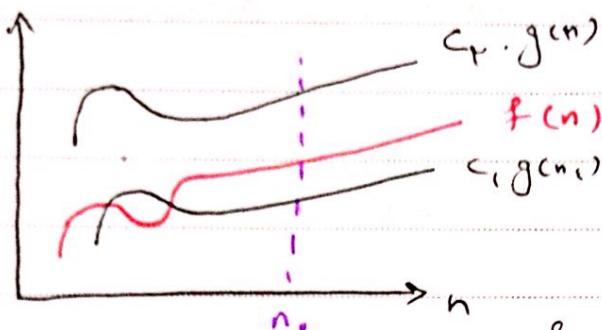
$$n \notin \Omega(\sqrt[n]{f(n)})$$

لذا $\sqrt[n]{f(n)}, n \sim 1, \Omega(\sqrt[n]{f(n)})$ نمایند

کامپیو

$$f(n) \in \Theta(g(n)) \iff \exists c_1, c_2 > 0, \exists n_0 > 0 \mid \forall n \geq n_0$$

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



$\Leftrightarrow f(n)$ بین $c_1 \cdot g(n)$ و $c_2 \cdot g(n)$ است، میانگین تابع $f(n)$ است.

$$\lim_{n \rightarrow \infty} \frac{g(n)}{f(n)} = c \neq 0 \quad n \in \Theta(n)$$

$$n \in \Theta(\ln n)$$

$$n \notin \Theta(n^2)$$

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

مبتداً می خواهد

$$T(n) \in \Theta(n^2)$$

برای این

$$T(n) \in O(n^2)$$

$$\in O(n^2)$$

$$f(n) + O(f(n)) \in \Theta(f(n))$$

مثال: \downarrow
کسر کوچک

9V, 7V

جنس چارم

دوتابع مثبت f, g

$$a, b \in \mathbb{R}$$

نحوه

$$f \in O(g) \quad \textcircled{1}$$

$$a < b \quad \textcircled{1}$$

$$f \in \Omega(g) \quad \textcircled{2}$$

$$a > b \quad \textcircled{2}$$

$$f \in \Theta(g) \quad \textcircled{3}$$

$$a = b \quad \textcircled{3}$$

حالات ممکن دوتابع f, g برای هر دوتابع f, g ممکن است

۱) تابع f بین g و $3g$ است

Subject:

Year Month. Day.

little o

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

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

$\Leftrightarrow f(n)$ از $g(n)$ کم

$$n + \alpha \in o(n^r)$$

$$n + \alpha \in o(n \cdot \log n)$$

$$T(n) \in o(n^r)$$

$$n + \alpha \notin o(n)$$

$$f(n) \in w(g(n)) \iff \forall c > 0 \exists n_0 \mid \forall n \geq n_0, f(n) > c \cdot g(n)$$

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

$\Leftrightarrow f(n)$ از $g(n)$ بیشتر

$$n + \alpha \in w(\sqrt{n})$$

$$n + \alpha \in w(\log n)$$

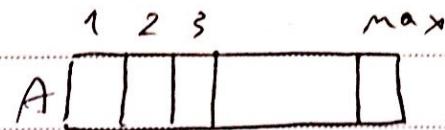
$$n + \alpha \notin w(n)$$

$$n + \alpha \notin w(n^r)$$

این کسی فایل دو کتاب و مجموعه است

$$f \in o(g) \& g \in o(h) \Rightarrow f \in o(h)$$
 خاصیت ترازوی

مسنونه ای فوار



: (Queue) is

FIFO

↓
First In

First Out

Add(Q, x)

Find(Q, x)

IsEmpty(Q)

IsFull(Q)

on/off

Subject:

Year. Month. Day.

queue { head (اولین عدد) ایضاً
tail (آخرین عدد) ایضاً

باید مسافت داشت از آن رایه استفاده کنیم.

نهایی درین ساده و بارگذاری استاد:

• in Add(Q, x)

1. if tail == max then

2. error ("queue is full!")

3. return

4. else

5. tail = tail + 1

6. Q[tail] = x

• in Del(Q)

1. if head > tail then

2. error ("queue is empty!")

3. return

4. else

5. x = Q[head]

6. head = head + 1

7. return x

• in Find(Q, x)

1. for i < head to tail do

2. if Q[i] == x then

3. return true

4. return false

YESTERDAY

On

07/07/

E

Subject:

Year. Month. Day.

٢

• \rightarrow IS EMPTY (Q)

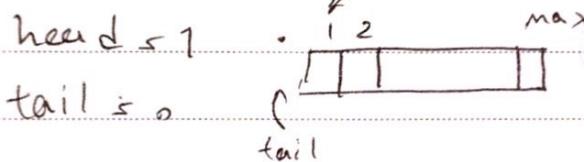
1. if head > tail then
2. return true
3. else return false

• \rightarrow IS FULL (Q)

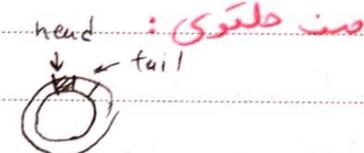
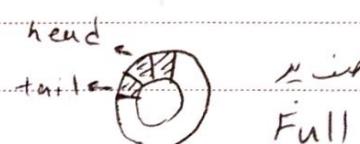
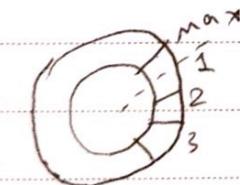
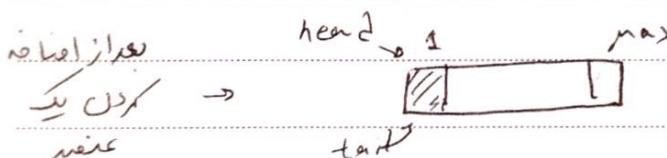
1. if tail == max then
2. return true
3. else return false.

Initialization

head



tail = 0



empty : head = tail

adds Q[tail] = x tail++

del: x = Q[head] head++

full: head < (tail + 1) ~~and max~~

cont

of Revision

- Add(Q, x)
0. if !ISFull(Q) then

1. Q[tail] = x

2. tail = tail + 1

3. if tail == max + 1 then

4. tail = 1

- Del(Q)
0. if !IsEmpty(Q) then

1. x = Q[head]

2. head = head + 1

3. if head == max

4. head = 1

5. return x

- ISFull(Q)

1. t = tail + 1

2. if t == max + 1 then

3. t = 1

4. if t == head then

5. return true

6. else return false

- ISEmpty(Q)

1. if tail == head then

2. return true

3. else return false

outfit

Subject:

Year Month. Day.

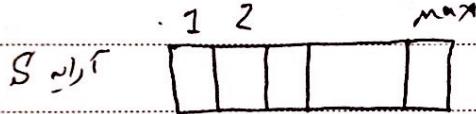
• Find(Q, x)

1. i = head
2. while ($i \neq \text{tail}$) do
3. if $Q[i] = x$ then
4. return true
5. $i = i + 1$
6. if $i = \text{max} + 1$ then
7. $i = 1$
8. return false

: (stack) ساخته (Create)

last In , First out \leftarrow LIFO مدل

برای پیاده سازی آن از تراکم استفاده کنیم



ویژگی های Stack push \leftarrow *

pop \leftarrow * Stack خالی شدن

IsEmpty \leftarrow *

ISFull \leftarrow *

Top \leftarrow Stack اولین عضو Stack

Stack \leftarrow اولین Top \leftarrow

top = 0 \leftarrow اولین عضو Stack

و شرط فرود \leftarrow

top = max \leftarrow عرضی کردن

07/07

Subject:

Year. Month. Day.

• IsEmpty(S)

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

• IsFull(S)

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

• push(S, x)

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

• pop(S)

1. if IsEmpty(S) then
2. error("stack is empty")
3. return
4. $x = S[\text{top}]$
5. $\text{top} \leftarrow \text{top} - 1$
6. return x

Q&A

2

Subject:

Year: Month: Day:



$x - a$



(-)

• Find(S, x)

1. for $i = 1$ to top do
2. if $S[i] == x$ then
3. return true
4. return false

برمجة

الآن \rightarrow تدريسي موسى - تدريسي موسى، كورس

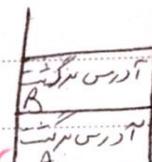
لذلك

function C() function B() function A()

call C()

call B()

return



برمجة

return

return

$A \rightarrow B \rightarrow C$

$(a+b) * c$
أيام رارو

$a + (b * c)$

برمجة

right مسiple infix فرمطاني $a + b * c$

idle مسiple postfix فرمطاني $a(b.c * +)$

مسiple right prefix فرمطاني $+ a * b c$

أدب

* تدريسي بطيء و معمولى (زى تانى باين استعمال)

الآن (زى تانى) تنبيل فهم صانونه بيسونا تانى نزا استعمال

• $\frac{d}{a} + (e * f) / g - b * c / d + e * (f - h)$: حل

مسقط

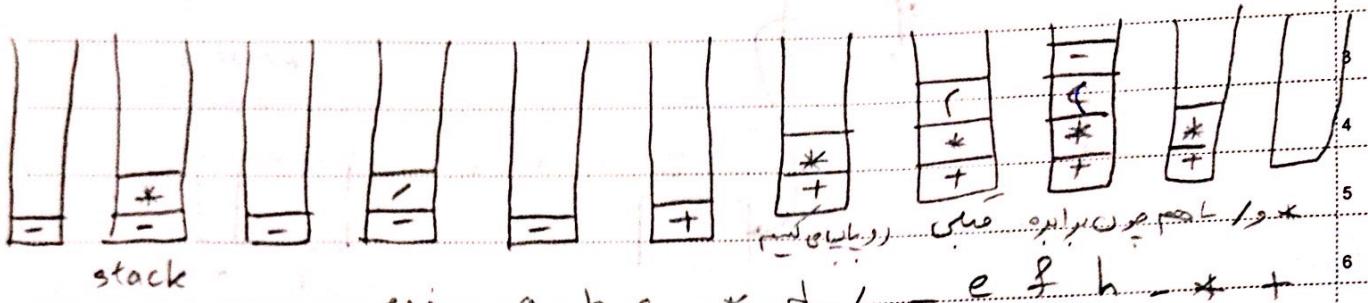
مسقط

مسقط

مسقط

مسقط

۱۲) آنچه ممکن است در عبارت $a + b * c - d / e ^ f + g$ در پیش از push خود را در



* پیش از push اولین سیگنال را در پیش push می کنیم \rightarrow روی براحتی باقی مانده push داشته باشیم

** فقط براحتی بسته هست هر چیزی که در stack باشد pop کنیم تا براحتی باز برسیم

۱۳) آنچه ممکن است در عبارت $a + b * c - d / e ^ f + g$ اولین سیگنال را در پیش از push داشته باشد pop شود. و در خود حذف شود.

۱۴) براحتی باز همیشه در پیش push ممکن است $a + b * c - d / e ^ f + g$ باشد تمام علائم را در

با براحتی باز برسیم pop شود و در خود حذف شود.

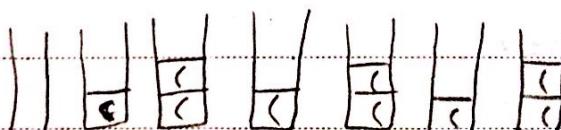
۹۷, ۷, ۹

جلد ششم

۱۵) در درسی درست براحتی لگاری از این استاده می کنیم.

۱۶) همان‌طورهایی بسته push می کنیم. اگر براحتی دیدیم که براحتی قبل روی pop می کنیم

$((a + b) * (c))$



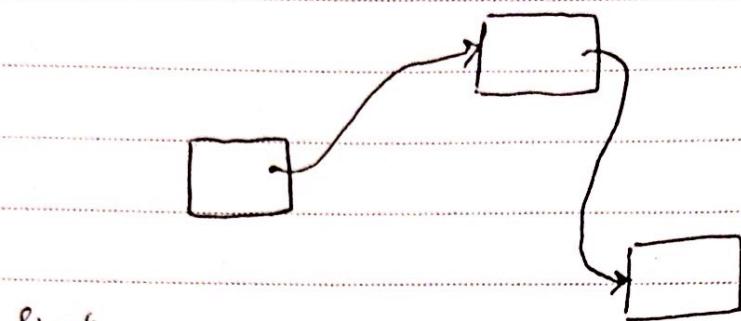
مثال

۱۷)

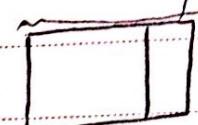
Subject:

Year. Month. Day.

فرستینی (Linked lists)

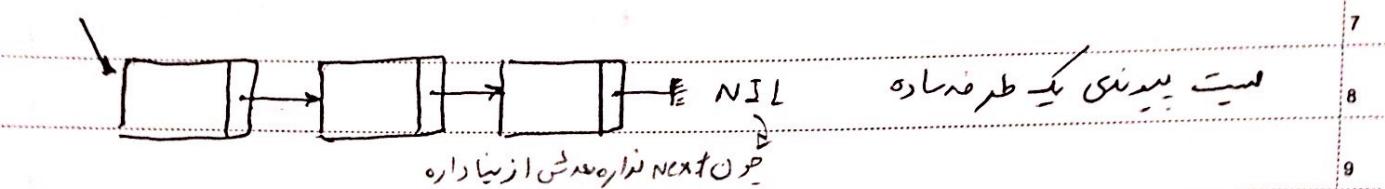


Record
Node

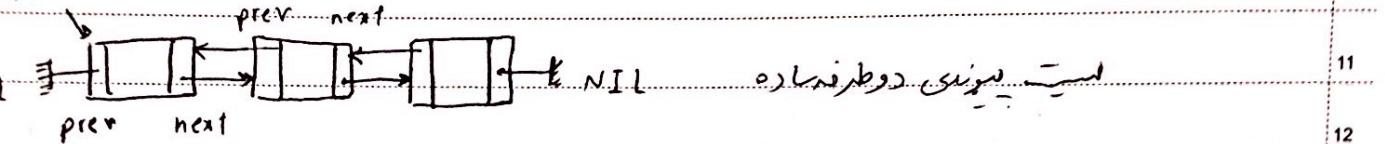


first

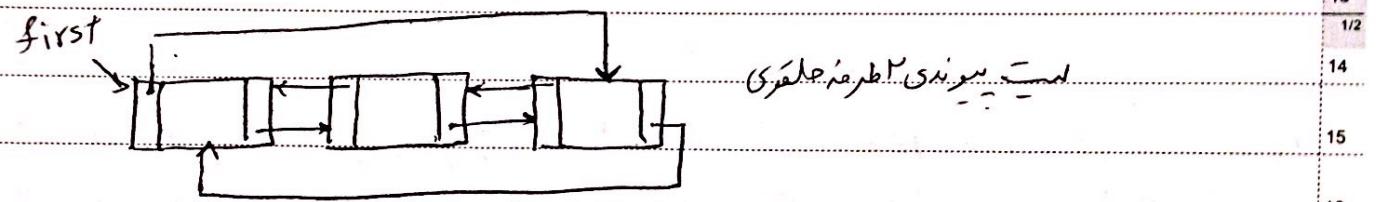
لیست پیوندی که طرفهای اطلاعاتی را در یک مکان ذخیره می‌کند



first



first



نکته: first گره اولین آدرس شروع اولین

نکته: first گره اولیه است \leftarrow NIL *

لیست پیوندی یک طرفه ساده

struct Node

{

int info;

node * next;

}

07/07/2023

Subject: _____
Year: _____ Month: _____ Day: _____

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26

```
Node * getNode()
```

{

Node * p;

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

return p;

free(ptr);

```
void main()
```

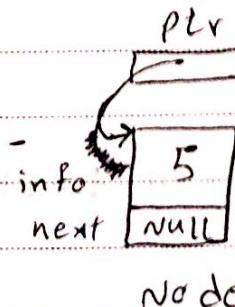
{

Node * ptr;

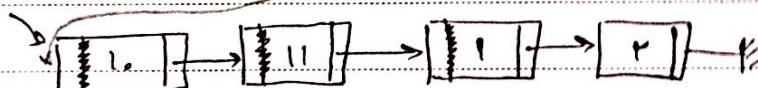
ptr = getNode();

ptr->info = 5;

ptr->next = NULL;



first → 10 11



پیاسیں لیت:

```
Node * ptr = first;
```

```
while (ptr)
```

{

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

```
ptr = ptr->next;
```

}

info info & first
info, last first
*first ← first
first →

out

A

Subject:

Year. Month. Day.

درج در لیست پسندی:

Node *p = getNode();

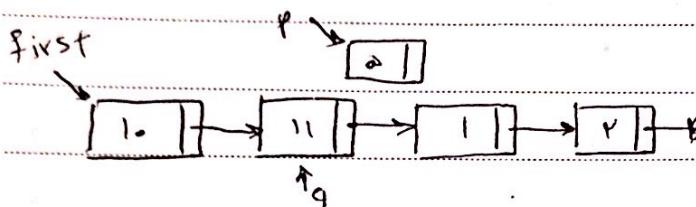
p->info = b;

p->next = first;

first = p;

شكل اخراج هدف ابتدای لیست

مثال: درج در بعداز 11



node *q = first;

while(q)

{

if (q->info == 11) break;

q = q->next;

}

if(q){

p->next = q->next;

p->next = p;

}

خط درود اول:

node *p = first;

if(first)

{

first = first->next;

free(p);

}

آخر

Subject:

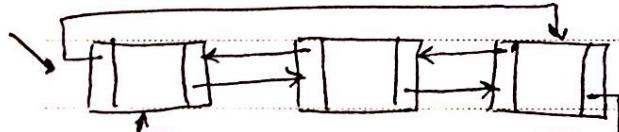
Year. Month. Day.

کارهای دفترچه ای میان راه است

```
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);
    }
}
```

✓✓✓✓✓

first



نیز خالی first = NULL

ست سرندی دو طرفه دلخواهی:

ویلچاریا

کار

۱

Subject:

Year Month Day.

یہاں لیتے:

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

: اپنے

Find(first, x)

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

: add

node *p = get Node();

p.info = 5;

node *p1, l1

Add(first, p)

1. if first == NULL then

exit

Subject:

Year. Month. Day.

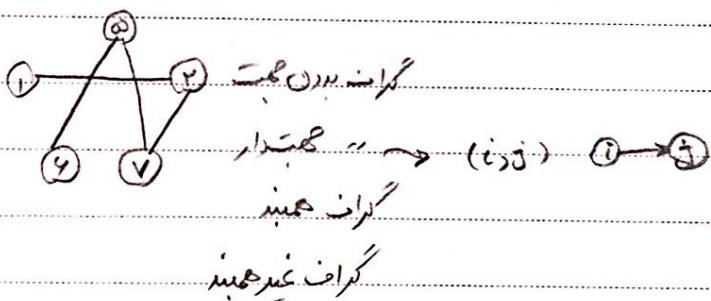
2. $\text{first} = p$
3. $p.\text{next} = p$
4. $p.\text{prev} = p$
5. else
6. $p.\text{next} = \text{first}$
7. $q = \text{first}.\text{prev}$
8. $q.\text{next} = p$
9. $\text{first}.\text{prev} = p$
10. $p.\text{prev} = q$
11. $\text{first} = p$

: (Tree) درخت ساخته ای

$$G = (V, E)$$

مکانیزم
vertix
edge

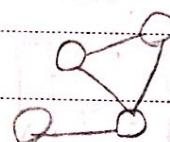
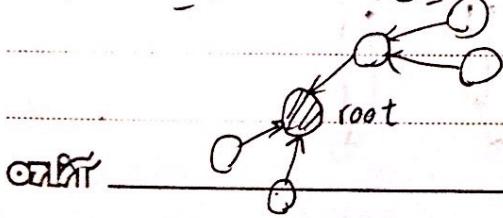
$$E = \{(1, 2), (2, 3), (3, 1), (1, 4), (1, 5)\}$$



= میریہ طبقہ cycle

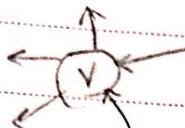
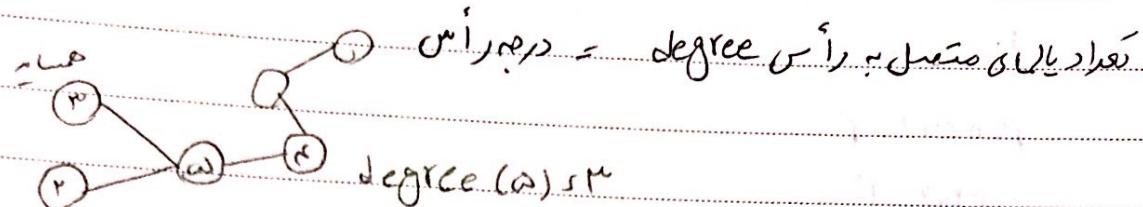
دراستہ میں بون حلقة = درخت آزاد

درخت ریتھ دار یہ کو خاص بام ریٹھ دار و لیکٹ سال ھائی سکھ ریٹھ کے



Subject:

Year. Month. Day.

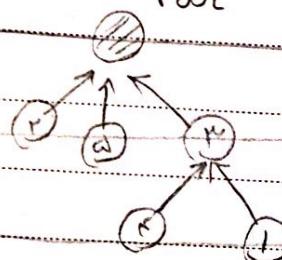


indegree(v) = 0

outdegree(v) = 3

degree(v) = 3

root



✓

✓ $v \in V$: indegree(v) = 1 \rightarrow Binary Tree

root

↓

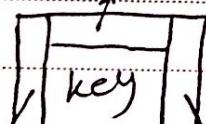
↙ ↘

parent
child

left
child

right
child

parent



: Binary Tree ساختار داده

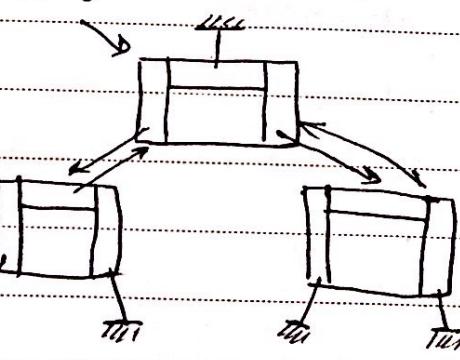
root

left
child

right
child

indegree(v) = 1

indegree(v) = 1



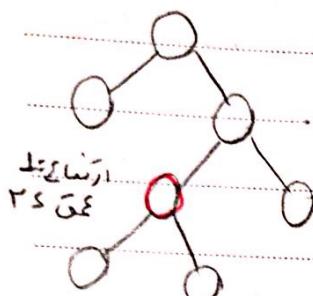
cont

ارتفاع کرمه $h(v) \leftarrow$ طول طولانیترین مسیر از فرزندان تا کرمه

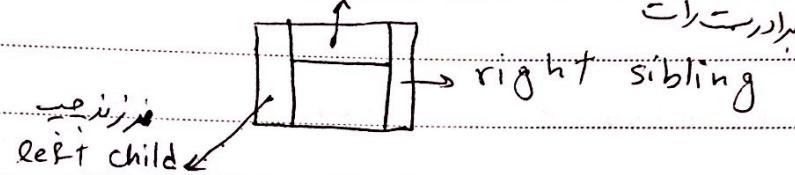
عن کرمه \leftarrow طول مسیر از ریشه تا کرمه

طول مسیر \leftarrow تعداد علایم های مسیر

ارتفاع درخت $h(T) \leftarrow$ ارتفاع ریشه درخت



عن کرمه



عن کرمه

برادر سمت راست

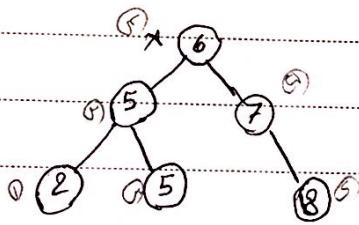
فرزند سمت چپ

۱۷/۱۴

$\Theta(n)$ space

جذب هشتم

* Binary search TREE (BST)



ویرگن درخت جستجوی دودویی:

اگر x کرمه درخت و y دیگر درخت بیس خواسته:

$y.key > x.key$ باید y در درخت لاست باشد و $y.key < x.key$

لیست مرتب کردهایی در درخت \rightarrow x میباشد

۱۶. جمله inorder $\rightarrow 2, 5, 2, 4, 5, 1$

درخت

۱۷. جمله $5, 7, 6, 8$

Inorder_Tree_Walk(x)

۱. if $x \neq \text{NULL}$ then

2. Inorder_Tree_Walk($x.left$)

$\Theta(n)$ حجمی

3. print $x.key$

4. Inorder_Tree_Walk($x.right$)

ابزارهای

۰۷/۸۵

۱۱

Subject:

Year Month Day

Sub
Yea

1 item

2 time

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

$T(0) = C$ $T(n) = O(n)$ \rightarrow $T(n) = O(n)$

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

? $T(n) = O(n)$ \rightarrow \times

پس $T(n) \leq (c+d)n + c$

کلی $n > 0$

$T(0) \leq (c+d) \times 0 + c$ $T(0) \leq c$ \checkmark

استنادی $n > 0$ برای

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

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

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

کلی $n > 0$

search(x, k) \rightarrow بارگذشت

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

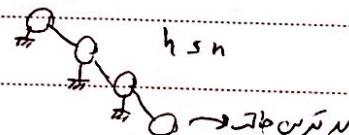
2. return x

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

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

5. else return search($x.\text{right}$, k)

O_b زمان $O(h)$ \rightarrow ارجاع درست h



logn

بین میانی ها تا درستی دستیابی

07/07/2023

Subject:

Year. Month. Date. ()

Iterative_Search(x, k)

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

$O(h)$

گستاخی کیا جائے کہ اسی طرز سے سچے بحث کی کسی نہیں کرتے۔
اسی طرز سے سچے بحث کی کسی نہیں کرتے۔

Tree_Min(x)

1. if $x = \text{NULL}$
2. $x.\text{left} = \text{NULL}$ then
3. return x
4. return Tree_Min($x.\text{left}$)

Tree_Min(x)

0. if $x = \text{NULL}$ return x
1. while $x.\text{left} \neq \text{NULL}$ do
2. $x = x.\text{left}$
3. return x

کوئی right child نہیں تھا تو x کو max ہے۔

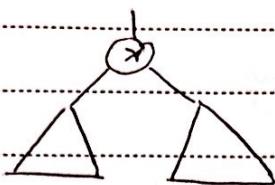
Tree_Successor(x)

1. if $x.\text{right} \neq \text{NULL}$ then
2. return Tree_Min($x.\text{right}$)
3. $y = x.\text{right}$
4. while $y \neq \text{NULL}$ and $x = y.\text{right}$
5. OVER $x = y$
6. $y = y.\text{left}$
7. return y

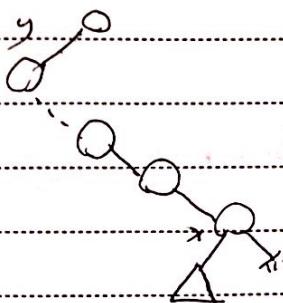
Subject:

Year. Month. Date. ()

8



x.right=NULL: ایجاد



while y!=NULL and x==y.right
x=x.parent

برای بحث برخوبی میکاریم

return y

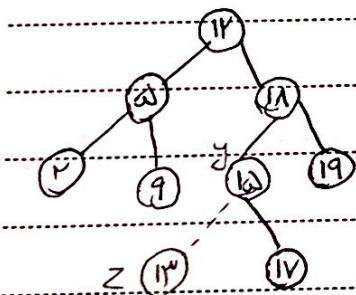
parent=p

برای یافتن یک جای جایگزین نویم (لطفاً ها و صین و میاکس خابه عکس)

qV, V, x1

حل فرم

Insert: که باید اینجا search بالای سریع باشد و درج کنیم



* نیز لست را در جای خود قرار دهیم

z.key <= 14

z.left=NULL

z.right=NULL

OVER

Subject:

Year. Month. Date. ()

Tree - Insert(T, z)

1. y = NULL

2. x = T.root

3. while x ≠ NULL do

 4. y = x

 5. if z.key < x.key then O(h)

 6. x = x.left

 7. else x = x.right

 8. Z = p(y)

 9. if z.key < y.key then

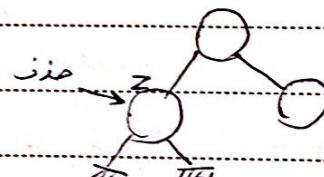
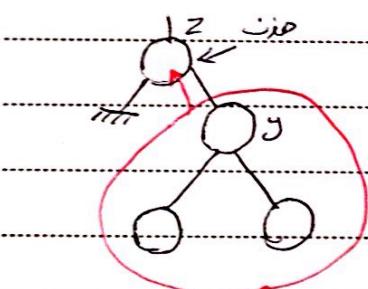
 10. y.left = z

 11. else y.right = z

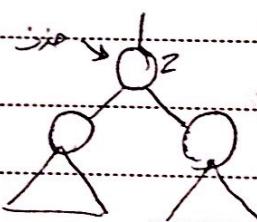
Delete:

حذف ممكناً ممكناً

حذف ممكناً ممكناً



z يساوي z يساوي



OVER

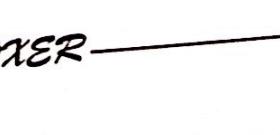
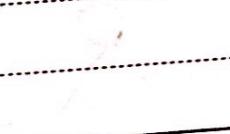
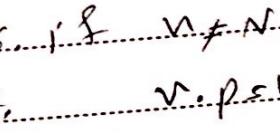
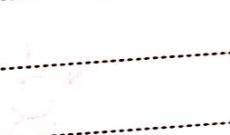
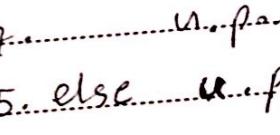
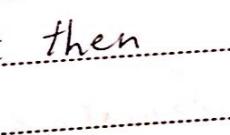
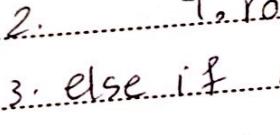
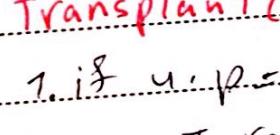
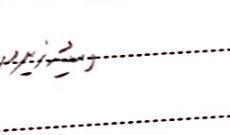
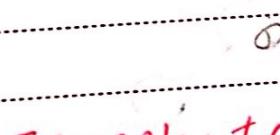
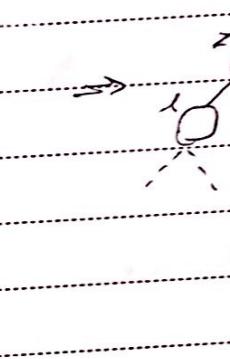
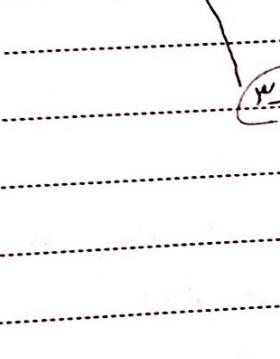
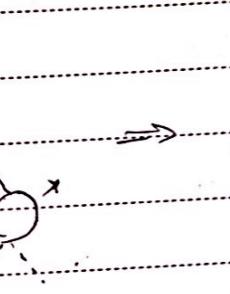
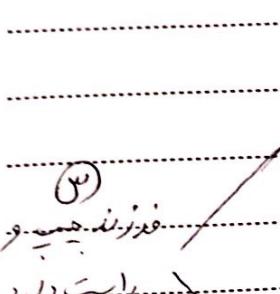
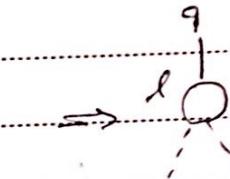
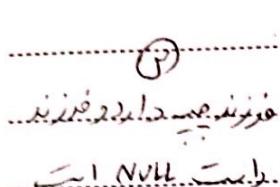
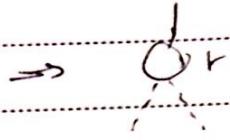
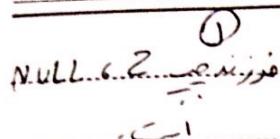
15

Subject:

Year.

Month.

Date.



Transplant(T, u, v)

1. if $u.p = \text{NULL}$ 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. if $v \neq \text{NULL}$ then

7. $v.p = u.p$

OVER

Tree - Delete (T, z)

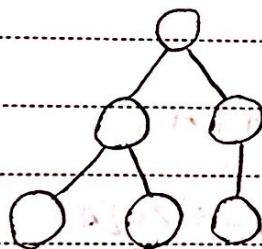
```

1. if  $z.left = \text{NULL}$  then
2.     Transplant( $T, z, z.right$ )
3. else if  $z.right = \text{NULL}$  then
4.     Transplant( $T, z, z.left$ )
5. else
    y = Tree-Min( $z.right$ )
    if  $y.p \neq z$  then
        Transplant( $T, y, y.right$ )
     $y.right = z.right$ 
     $y.right.p = y$ 
    Transplant( $T, z, y$ )
     $y.left = z.left$ 
     $y.left.p = y$ 

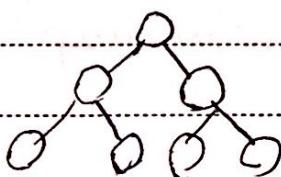
```

AV, V, RR

Heap Circular



Complete (Binary) Tree



不完備二元樹 (Incomplete Binary Tree)

OVER

is perfect if $V_{j+1} \leq V_j$, it is not perfect.

3. t

4. e

5. a

6. a

7. e

8. e

9. e

10. e

11. e

12. e

13. e

14. e

15. e

16. e

17. e

18. e

19. e

20. e

21. e

22. e

23. e

24. e

25. e

26. e

27. e

28. e

29. e

30. e

31. e

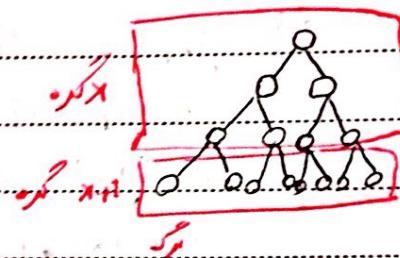
32. e

..... is Max Heap.

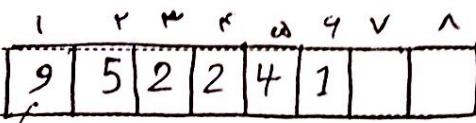
maximum: Max Heap $\rightarrow O(1)$

minimum: Min Heap

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



از بین max heap

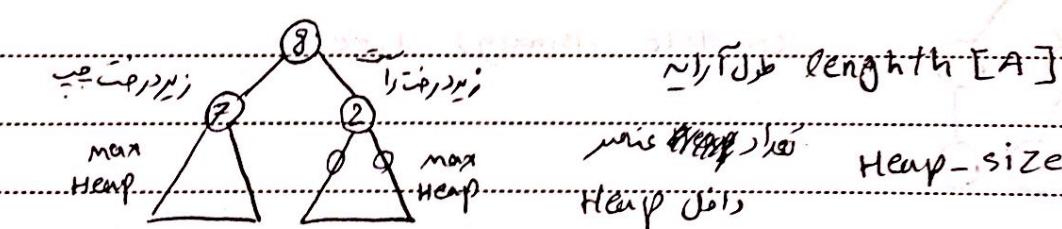


i. \rightarrow $\lfloor \frac{i}{2} \rfloor$

parent(i) $\rightarrow \lfloor \frac{i}{2} \rfloor$

left(i) $\rightarrow 2i$

right(i) $\rightarrow 2i + 1$



length[A] \rightarrow max heap \rightarrow Heap-size[A]

in a max heap, if $A[i] < A[\text{left}(i)]$ or $A[i] < A[\text{right}(i)]$, then swap $A[i]$ with $A[\text{left}(i)]$ or $A[\text{right}(i)]$.

max-Heapify(A, i)

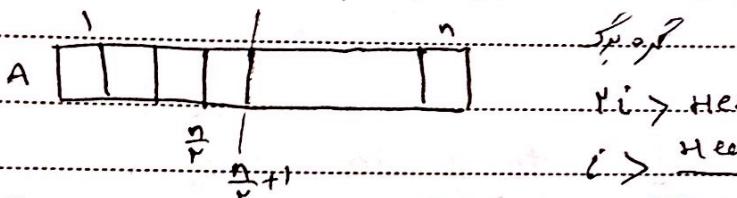
1. if $\text{left}(i) \leq \text{Heap-size}[A]$ and

$A[\text{left}(i)] > A[i]$ then

OVER

2. $\text{largest} = \text{left}(i)$
3. else $\text{largest} = i$
4. if $\text{right}(i) < \text{Heap_size}[A]$ and
 $A[\text{right}(i)] > A[\text{largest}]$ then
5. $\text{largest} = \text{right}(i)$
6. if $\text{largest} \neq i$ then $O(n) = O(\log n)$
7. $\text{swap}(i, \text{largest})$
8. $\text{max_Heapify}(A, \text{largest})$ $T(n) = T\left(\frac{n}{2}\right) + C = O(\log n)$

Build max heap first from max heap \rightarrow الخطوة الأولى



$i/2 > \text{Heap_size}[A]$

$i > \text{Heap_size}[A]$

نهاية المقدمة

Build_max_Heap(A, n)

$O(n)$ 1. for $i = \lceil \frac{n}{2} \rceil$ down to 1 do

$O(n)$ 2. $\text{max_Heapify}(A, i)$

لأن كل ترتيبات $O(n \cdot \log n)$

$O(n^2)$ tight

ويجب

ارتفاع $\log n$

$C \times \log n$

$\log n$

$$\text{لدينا: } C \times \frac{n}{r} \times 1 + C \times \frac{n}{r} \times r + \dots$$

$$C \times \frac{n}{r} \times r + \dots + C \times 1 \times \log_r n$$

$$\sum_{i=1}^{\log_r n} C \frac{n}{r^{i+1}} \times i = O(n)$$

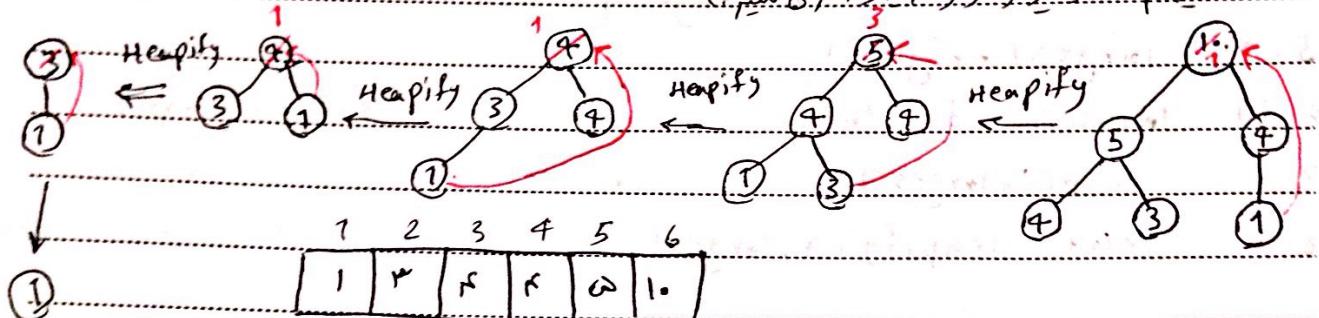
$$C \times 1 \times \frac{n}{r}$$

OVER

10

پڑھا
Heap sort

باین حالات کل کر ریسے را برداشت و درون آفرین جویند لذاریم و پس مسے اسے
آفرین و باین آفرین فرزند را استخراج کرد و به عبارتی ریسے کل کاریم و پس
آفرین (لشید هر مرتبه کل کاریم کیسے) (Heap ایسا)



Heap sort (A)

1. Build-Max-Heap(A) $O(n)$
2. for $i \leftarrow \text{length}[A]$ down to 2 do $c_p \times n$
3. swap($A[1], A[i]$) $c_w \times (n-1)$
4. heap-size[A]-- $c_r \times (n-1)$
5. Max-Heapify($A, 1$) $O(\log n) \times (n-1)$ ~~الوقت~~

$$\sum_{i=1}^{\infty} i \times \frac{x^i}{(1-x)^2} = O(n \log n) \rightarrow \text{فقط O(n log n)}$$

$1 \leq i \leq n$

$$5. \text{ time } \leq \sum_{i=1}^{n-1} \log i = \log \prod_{i=1}^{n-1} i = \log(n-1)! \approx \Theta(n \log n)$$

: Balanced Binary Search Tree

AVL Tree

✓ Red-Black Tree

2-3 Tree

B-Tree

OVER

: Red-Black Tree

درخت حسب جری دودویی که افرادی آن را نمی‌دانند. با قوی محدود است.

و در این درخت درجه ایست.

۱) هر درخت قدرت سیاه است.

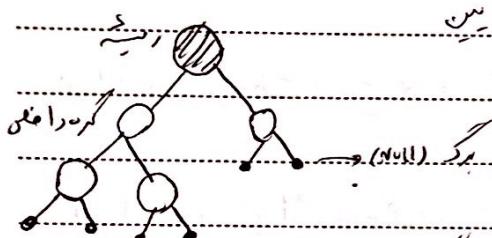
۲) درخت سیاه است.

۳) صورت (بالا) صیاه است.

۴) دکمه هر قدر می‌باشد. هر دکمه آن سیاه است.

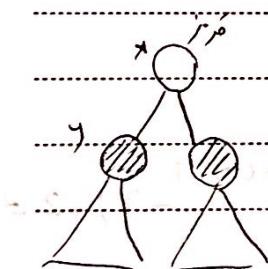
۵) دکمه هر کدام دهد مسیر های ساده از آن دکمه بگذرد. مقدار دمراهی سیاه مساوی دارد.

بین

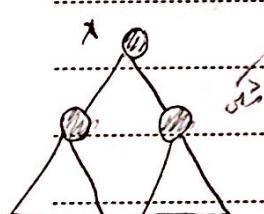


خط که مسیر از کامپیوئن که حداقل ۲ برابر طول که مسیر بگیرد از همان کامپیوئن بگیرد. دیگر است.

مقدار دمراهی سیاه در مسیر از بزرگ تا بزرگ ناچیل خود است. $bh(x)$ black height

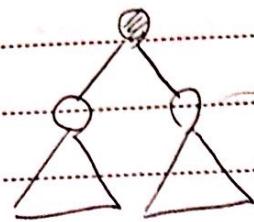


$$bh(x) = bh(y) + 1$$



$$bh(x) \leq bh(y)$$

OVER



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

برجایت کوچک

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

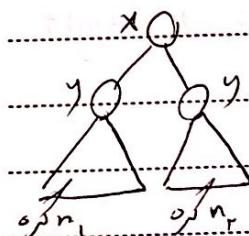
لگاریتمی

حکم: مود رفتہ جہاں داٹا لرینگ جد ائے Red - Black

ادعا: جہڑی مود رفتہ باریستہ جہاں 1. $2^{bh(x)}$ لر داٹا لر

این ادعا بآسانی اثبات ہے

(آنے والے فرینٹ) صفاں لے اندرون لرینگ سیدھا، $bh(y) \geq bh(x) - 1$



فرنٹ استدار

$$\begin{aligned} n &> 2^{bh(y)} - 1 \\ &\geq 2^{bh(x)-1} - 1 \end{aligned}$$

$$n \leq n_l + n_r + 1 \geq (2^{bh(x)-1} - 1) + (2^{bh(x)-1} - 1) + 2 =$$

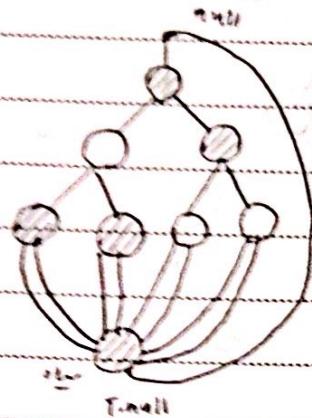
$$\Rightarrow 2^{\text{bh}(x)} - 1 \Rightarrow n \leq 2^{\text{bh}(x)} - 1$$

$$bh(x) \geq \frac{h(x)}{r} \quad (5)$$

$$Q, R \Rightarrow n \geq r^{bh(x)} - 1 \geq r^{\frac{h(x)}{r}} - 1$$

$$h(x) \leq \lg_r(n+1)$$

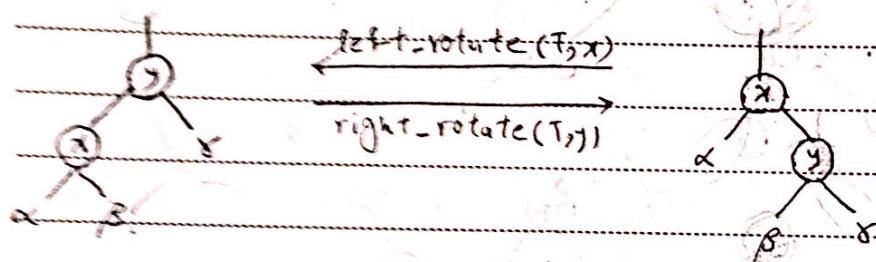
OVER

~~possible~~~~give address of T, null and only null (when tree is full)~~

Tnull

Tnull

: Rotation

left-rotate (T, x) $y.left = x$ 1. $y = x.right$

y.right = x

2. $x.right = y.left$

y.left = x

3. if $y.left \neq T\text{null}$ then

y.left = x

4. $y.left = x$

y.left = x

5. $y.p = x.p$

y.p = x.p

6. if $x.p = T\text{null}$ then

y.p = x.p

7. $T.root = y$

y.p = x.p

8. else if $x = x.p.left$ then

y.p = x.p

9. $x.p.left = y$

y.p = x.p

10. else $x.p.right = y$

y.p = x.p

11. ~~box~~ $y.left = x$

y.left = x

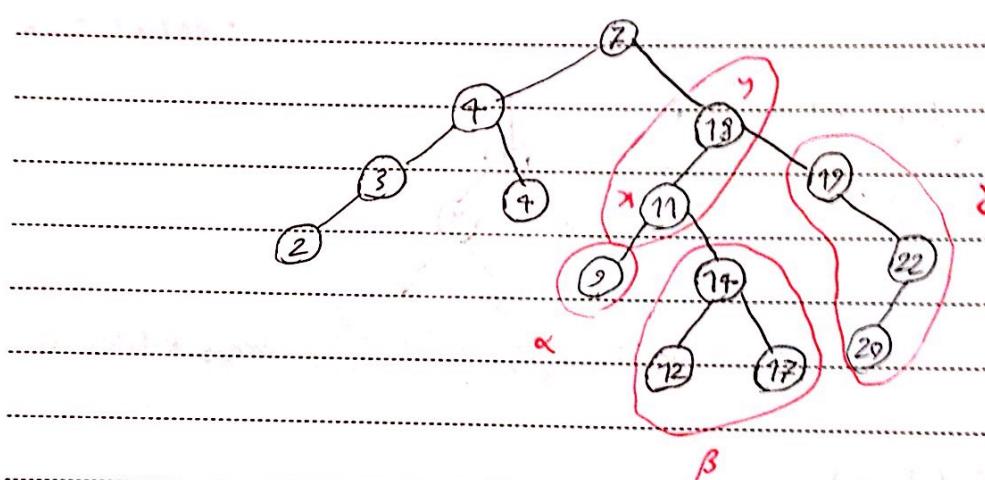
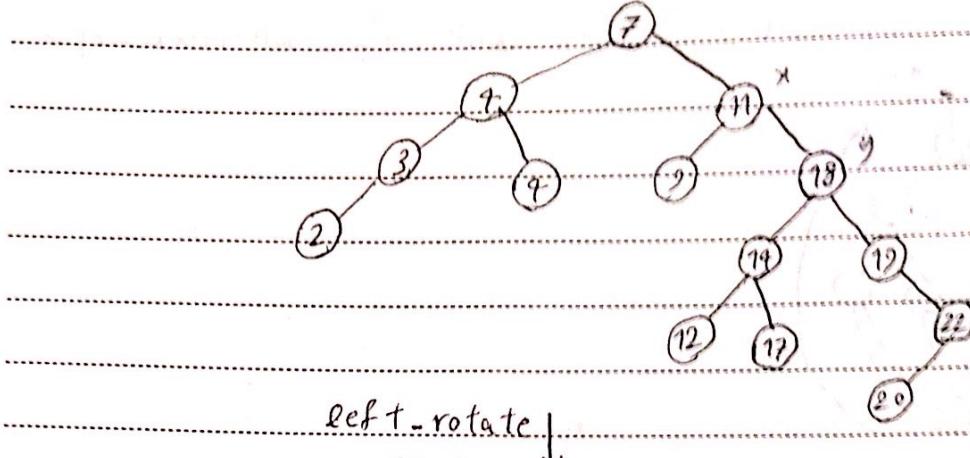
12. $x.p = y$

y.p = x.p

IV

Subject :

Year. Month. Date. ()



R.B.-Insert(T, z)

1. $y = T.\text{null}$

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

3. while $x \neq T.\text{null}$ 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 $y = T.\text{null}$ then

10. OVER $T.\text{root} = z$

1A

11. else if $z.key < y.key$ then

12. $y.left = z$

13. else $y.right = z$

14. $z.left = T.\text{null}$

15. $z.right = T.\text{null}$

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

RB-Insert-Fixup(T, z)

1. while $z.p.color \neq \text{RED}$ do

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

3. $y = z.p.p.right$
 if $y.color == \text{RED}$ then

 case } $z.p.color = \text{Black}$

 case } $y.color = \text{Black}$
 $z.p.p.color = \text{RED}$

$z = z.p.p$

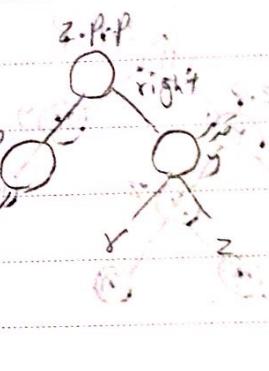
 else if $z == z.p.right$ then

$| z = z.p$

 left-rotate(T, z)

9V, 1, 0

black



RB-Insert(T, z)

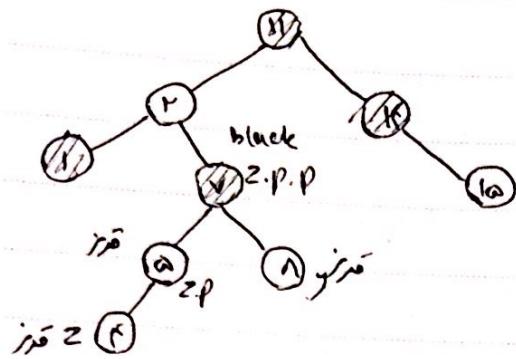
{ ; }

 RB-Insert-Fixup(T, z)

$O(\log n)$

P4PCO Fix up

$z.p = z.p.p.left$ فرزند میانی $z.p$



case 1:

$y = z.p.p.right$

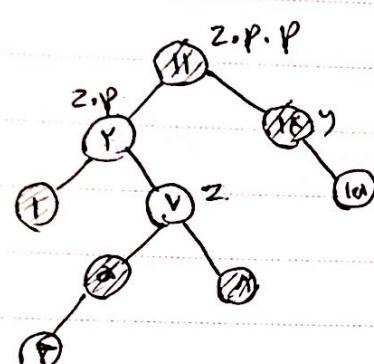
میانی راست

برای حل مشکل

و در $z.p$ سیاه شود

فرزند $z.p.p$ سیاه شود

نحو $z = z.p.p$,



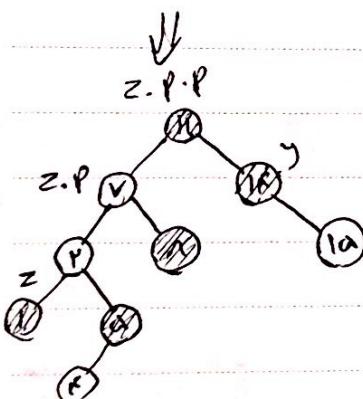
case 2:

$y = z.p.p.right$

میانی راست، ایستاده راست

: case 2 \Rightarrow case 3 بدل

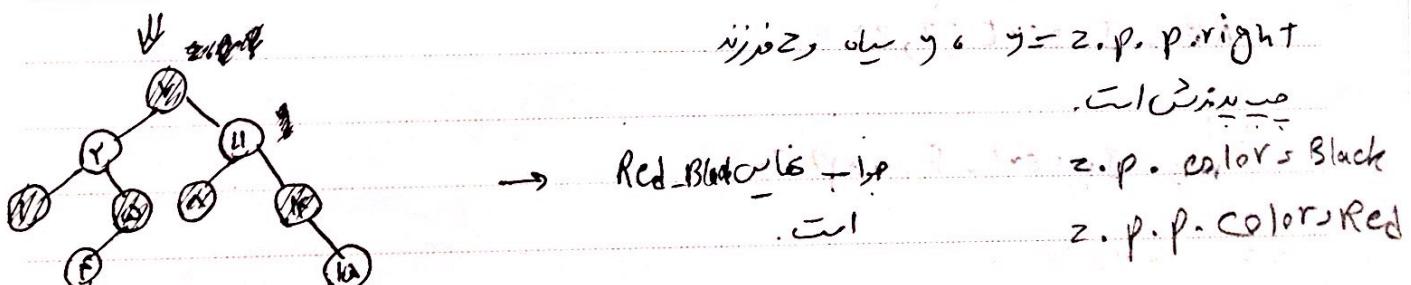
$z = z.p.left - \text{Rotate}(T, z)$



case 3:

میانی z و y هم راست $y = z.p.p.right$

میانی راست



\rightarrow Red-Blackout

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

ایستاده

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

PAPCO

ادامه

RB - Insert - Fixup (T, z)

1. while $z.p.color = \text{RED}$ do
2. if $z.p = z.p.p.left$ then
3. $y = z.p.p.right$.
4. if $y.color = \text{RED}$ then **case 1**
5. $z.p.color = \text{Black}$
6. $y.color = \text{Black}$
7. $z.p.p.color = \text{RED}$
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. $z.p.color = \text{Black}$
13. $z.p.p.color = \text{RED}$ } **case 3**
14. right-Rotate ($T, z.p.p$)
15. else (same as then --) \rightarrow right, right, left tree
16. $T.root.color = \text{Black}$

ثبت صدقه کردارای هر مرحله ای را در این loop invariant می‌دانیم.

گروه Z محفوظ است **a**

اگر $z.p$ رسیده باشد $T.z.p.p$ رسیده است **b**

اگر در پیش و پیشگاهی T حداقل یکی از دو دختر گر را رانتفون کند

و دو دختر گر **۱** و **۲** را رانتفون کند اگر دو دختر گر **۲** را رانتفون کند دلیلیست این است که z رسیده اس را محفوظ است. اگر دو دختر گر **۴** را رانتفون نکند دلیلیست این است که $z.p$, z , $z.p.p$ هر ۲ دختر گر محفوظ است.

بررسی محتوی درج در Red-Black

اشارات:



حالات اولیه (قبل از اولیه حلقه while)

درست است $z.p.color = RED$ و در اینجا هر آر داده شود.(۱) در اینجا در خصوص red-black، است یعنی اگر p ریشه باشد سیاه است.(۲) بعد از اضافه شدن z به درخت red-black و قرار دادن $z.color = RED$ و پس از آن دو بروزگشانی تراویح نفعن شود.

اگرچه ریشه درخت باشد و پس از ۳ بروزگشانی تراویح نفعن شود.

و پس از ۲ بروزگشانی تراویح نفعن شود.

و پس از ۲ بروزگشانی تراویح نفعن شود.

از اگرچه ریشه باشد یعنی اگر p ریشه باشد و پس از ۴ بروزگشانی تراویح نفعن شود.

حالات خاتمه (Termination)

while $z.p.color = RED$ doو پس از ۳ بروزگشانی تراویح نفعن شود پس از $z.p.color = RED$ سیاه است. اینها ممکن است در ریشه باشند.طبق (۱) رند ۲ کسر است. و پس از ۲ بروزگشانی حمره اگر p black باشد $p.root.color = root$. قرار گیرد و پس از ۲ بروزگشانی تراویح نفعن شود.

حالات نگهداری (maintenance)

توقفی کنیم (۱) و (۲) و (۳) در اینجا یک کاملاً حلقة while درست است. اشارات

ی که نیم کاملاً در اینجا یک کاملاً بدی حلقة نیز درست است

Subject:

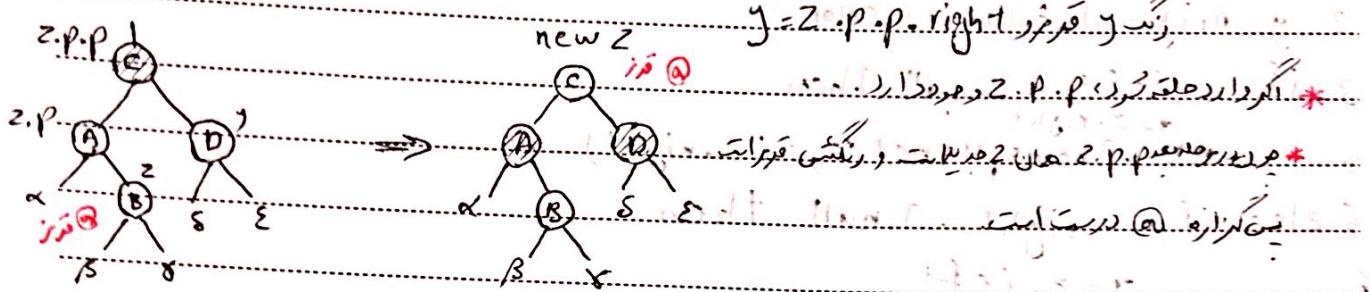
Year.

Month.

Date. ()

case 1: اثبات حالات ناممادی برای 1

case 1:

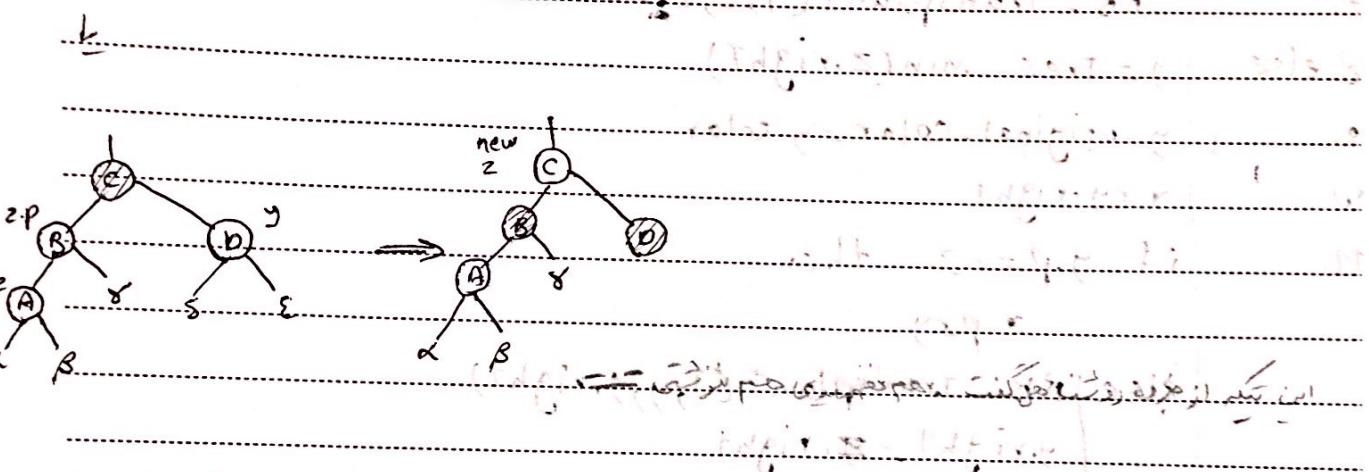


$$y = Z \cdot P \cdot p \cdot \text{right}$$

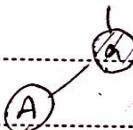
البروتوكول للحالة 1 هو $Z \cdot P \cdot p \cdot \text{right}$

* حالت 2 هي معاكسة حالت 1، حيث $Z \cdot P \cdot p \cdot \text{left}$

بروتوكول للحالة 2 هو $Z \cdot P \cdot p \cdot \text{left}$



case 2:



9V/11/12

RB-Transplant (T, α, ν)

جسی پارزیت

عل خود در جریان Red-Black tree

1. if $\alpha.p = T.\text{null}$ Then

2. if $T.\text{root} = \nu$

3. else if $\nu = \alpha.p.\text{left}$

4. $\alpha.p.\text{left} = \nu$

5. $\alpha.p.\text{right} = \nu$

6. $\nu.p = \alpha.p$

Subject:

Year. Month. Date. ()

RB_Delete(T, z)

1. $y = z$

2. $y.\text{original_color} = y.\text{color}$

3. if $z.\text{left} == T.\text{null}$ Then

$x = z.\text{right}$

4. RB_Transplant(T, z, z.right)

5. else if $z.\text{right} == T.\text{null}$ then

6. $x = z.\text{left}$

7. RB_Transplant(T, z, z.left)

8. else $y = \text{Tree_min}(z.\text{right})$

9. $y.\text{original_color} = y.\text{color}$

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

11. if $y.\text{p} == z$ then

$x.\text{p} = y$

else | RB_Transplant(T, y, y.right)

$y.\text{right} = z.\text{right}$

$y.\text{right}.p = y$

RB_Transplant(T, z, y)

$y.\text{left} = z.\text{left}$

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

$y.\text{color} = z.\text{color}$

if $y.\text{original_color} == \text{Black}$ Then

RB_Delete_Fixup(T, x)

حکم اسے ایسا دو : $y.\text{original_color} == \text{Black}$

و غیرہ کو خر آن حاکمین آن درج ہے

(۱) نعم ← خر اسے $x.p = x.p$, $x.p = x$

(۲) درج خر و بیشترین راستہ اسے $x.p = x$

OVER

RB_Delete_Fixup(T, x)

```

1. while x ≠ T.root and x.color = Black do
2.   if x = x.p.left then
3.     w = x.p.right
4.     if w.color = Red then
5.       w.color = Black
6.       x.p.color = Red
7.       left-rotate(T, x.p)
8.     w = x.p.right
9.     if w.left.color = Black and w.right.color = Black then
10.    w.color = Red
11.    x = x.p
12.    else if w.right.color = Black
13.      w.left.color = Black
14.      w.color = Red
15.      Right-rotate(T, w)
16.      w = x.p.right
17.      if w.color = x.p.color
18.        w.p.color = Black
19.        w.right.color = Black
20.        left-rotate(T, x.p)
21.        x = T.root
22.    else (same as then left → right)
23.  x.color = Black

```

case 1 case 2 case 3 case 4

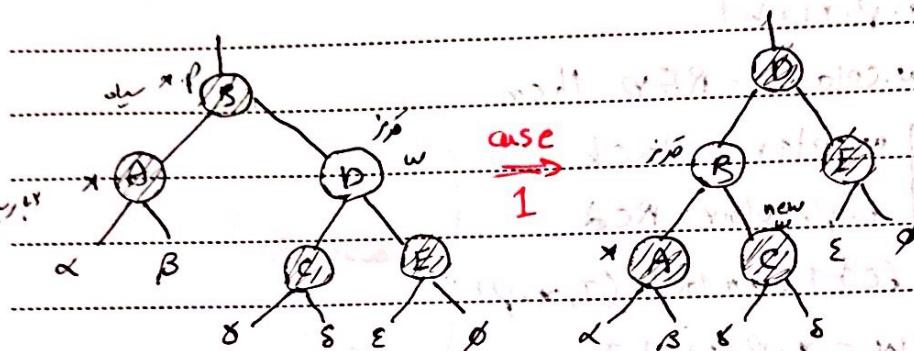
OVER

Subject: _____
Year. _____ Month. _____ Date. () _____

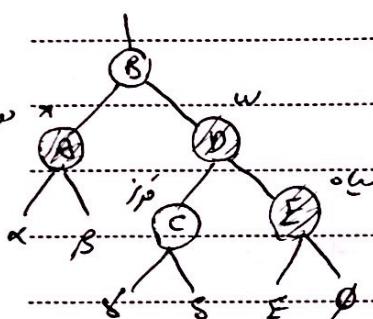
case 1

w.sx.p.right

w.colors.Red



case 3:



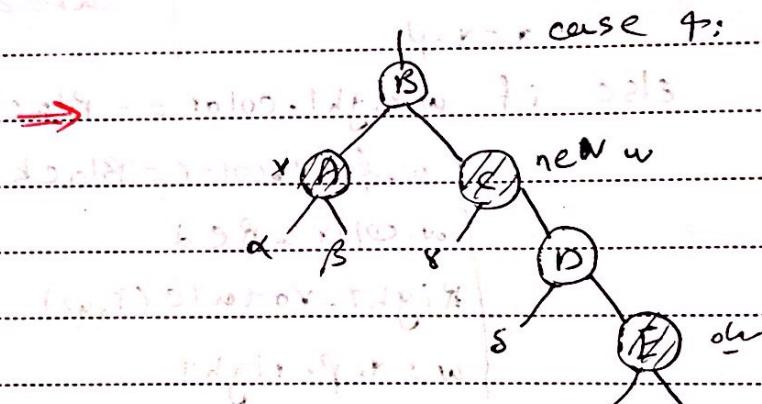
w.sx.p.right

w.colors.red

w.left.colors.ip

w.right.colors.ip

case 4:



OVER

(Data structure) Augmenting

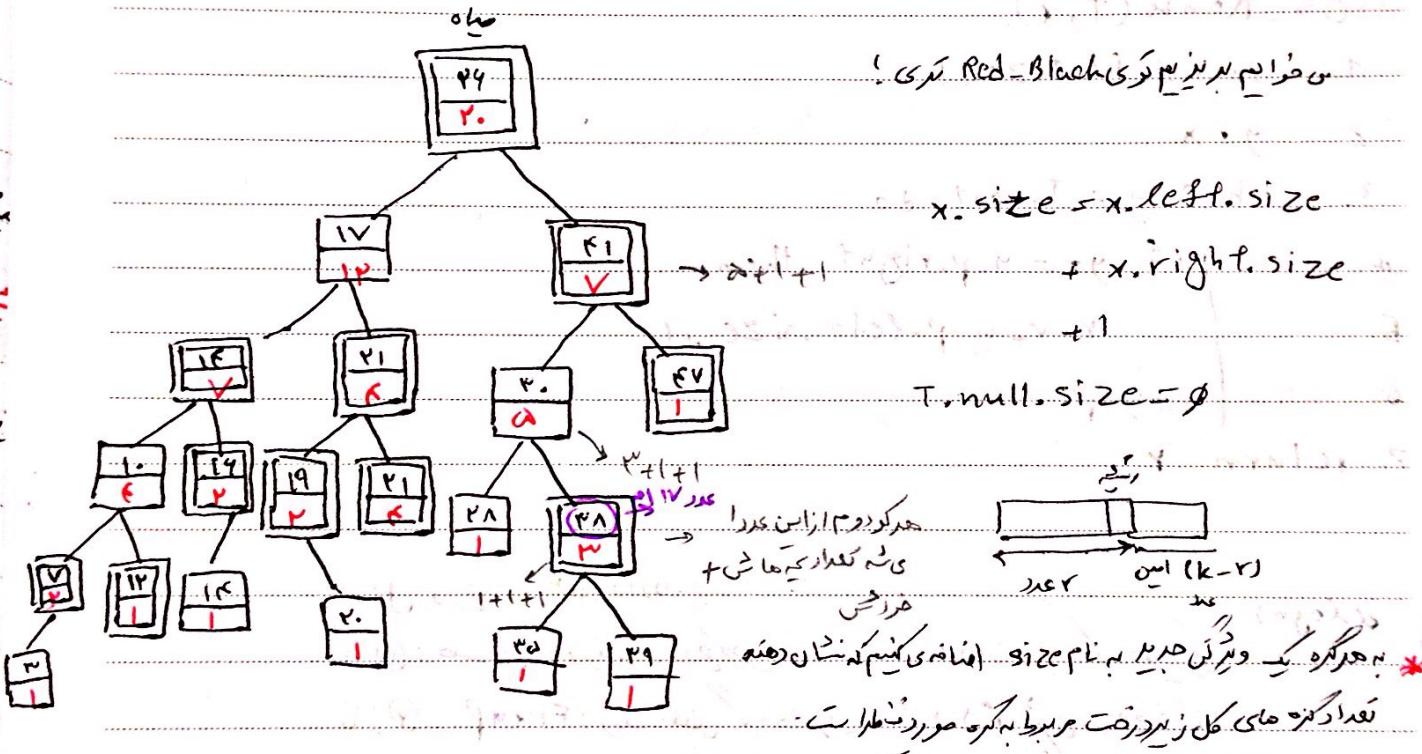
پرسنی اطلاعاتی را با ساختن داده های سرچر اضافه کنید که برآش مسائل ستری را با حل داده ای حل کنیم

ساختن داده های سرچر را Red-Black Tree نامید

($\frac{n \ln n}{k}$) \rightarrow کسری این داده های سرچر اضافه کنید که بحث کنیم که برآش مسائل ستری را با حل داده ای حل کنیم

Rank \rightarrow عددی که در درخت رتبه درخت است (ورودی: عده ها)

هر ۲ با مرتب سازی رزیکی $n \log n$ حل شود



حل مسائل Red-Black Tree : insertion, selection

نیمان : $O(n \log n)$

query نیمان : $O(\log n)$

kstV \rightarrow ۳۵۹۱۰ \rightarrow ۳۵۸۱۰ \rightarrow ۳۵۷۱۰ \rightarrow ۳۵۶۱۰

kstV

۳۵۸۱۰ \rightarrow ۳۵۷۱۰ \rightarrow ۳۵۶۱۰

kstV - ۱

۳۵۷۱۰ \rightarrow ۳۵۶۱۰

Subject

Date

برگه اول

OS-select(x, k)

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

2. if $k=r$ then

3. return x

4. else if $k < r$, then return OS-select($x.\text{left}, k$)

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

OS-Rank(T, x)

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

2. $y = x$

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

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

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

6. $y = y.\text{p}$

7. return r .

: Ranking دلخواه

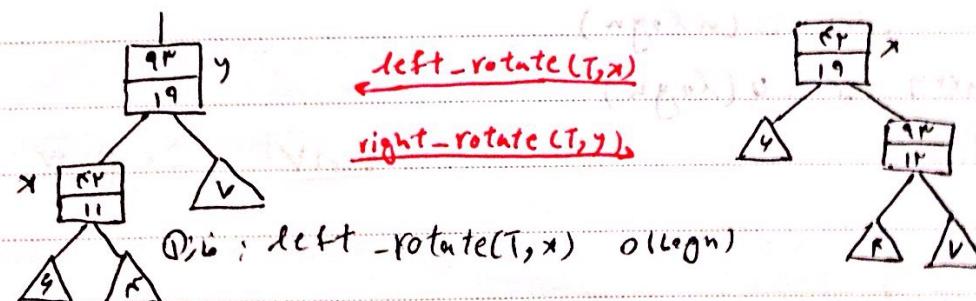
: size دلخواه

در عمل درج، ۲ فاز تارید.

فاز ①: مستعد در درخت درج کردن، بسایز هر گرهها از رشته تابعه حدید یک اثباتی تحریر

فاز ②: رنگ سینی گره ها عرضی تحریر (گاه شیری را size نیز نویسند)

و عملیاتی عجل ایجاد شوند.



و عملیاتی عجل ایجاد شوند.

$O(1)$; $\text{left-rotate}(T, k)$ در انتهای عمل

$\text{y.size} = x.\text{size}$ $O(1)$

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

دستور حذف: ۲ مازدادر

حذف: عملیات حذف: از size صفر ها از ریشه تا محل اولین گره (که حاکم زیرین نبود) حذف

نمایش: $O(\log n)$ میم خود

Fixup: $O(1)$ ریشه یعنی گره های غرفتی در ریشه سریع نمایند

rotate: دفعه دار بازیابی انجام خواهد

۴۷, ۸, ۱۹

Interval Tree

i بازه Interval

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

$$[t_l, t_r]$$

$$i.\text{low} = t_l$$

$$i.\text{high} = t_r$$

۲ بازه i و i' حالت است به هم دارند:

(۱) i, i' همیشان (overlap) ندارند.

\rightarrow overlap بین i و i' نباشد.

(۲) i' است جیب i بابت.

(۳) i' نسبت راست i' است.

x.int: اطلاع از دردست کرده بازه را بعرفتی کند.

این Red-Black Interval Tree همان ساختار درخت باین سریع است.

Interval_Insert(T, x) ①

Interval_Delete(T, x) ②

گروهی در درختی اینجا چو چو با اشتراک دارد.

Interval_Search(T, i) ③

PAPCO

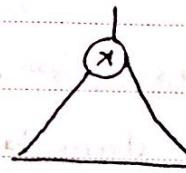
۲۳

Subject _____

Date _____

کلید هر لوره حدیثیں بازه است $x.\text{int_low} = x.\text{key}$

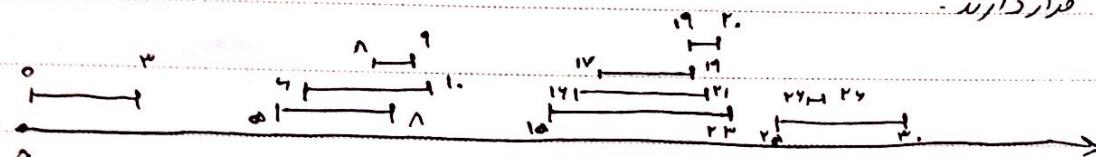
اطلاع افزوده $\leftarrow x.\text{max}$



مقدار بینی حد بازه های

گره های درخت بارگذاری شده

قدار دارند.



برای red-black tree برای اینها توی

Tree



int

ستک ها \leftarrow مطابق!

max

[19]

22

[20]

20

[23]

23

[23]

23

[19, 21]

20

[27, 29]

28

[23]

23

[4, 10]

10

[19, 20]

20

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

$\rightarrow (x, t) + \text{local_boundary}$

$\rightarrow (x, t) + \text{global_boundary}$

$\rightarrow (x, t) + \text{global_boundary}$

PAPCO

Interval Search (T, i)

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

search $[rr, ra]$ search $[l, l^r]$

گزاری اطلاعات در $O(\log n)$ انجام می‌شود

* تابع صفر: آندرخت T شامل کی از زیرشدن بازه‌ی i حیثیت‌شان دارد، آنکه زیرخت باریش x شامل چنین بازه‌ای است

(حالات) Initialization.

در این‌جا حالات تابع صفر $x = T.root$, i هست

در این‌حالات تابع صفر به طور بدیکه برقرار است. چن T همان زیرخت باریش x است

(تکراری) Maintenance.

آندر استای i کام مخفی تابع صفر درست است. آندر استای i کام مخفی مخفی تابع صفر درست است

حالت درستی باشد.

i^r, i^l

آنقدرست else ابرآشوند (خطه) گردد:

$i^r.high < i^l.high - x.left.max < i^l.low$

بازه‌های متضمن است چه استراکت دارد



بسیار بازدیدی در آن وجود دارد که نه استراکت دارد.

آن بازده در زیرخت $x.right$ است.

Subject

Date

آموزه درست نباید در کل درست باشد.

اگر $i < i'$ بود (فقط) آنگاه عکس نتیجه نمایش حلقه برقرار است .
در این حالت :

اگر $i > i'$ باشد آنگاه صورت مغلوب است .

$i > i'$ نیز دارای نتیجه مغلوب است .

برای هر i در زیر درخت راست :

$i < i' < i'' < i''' < \dots < i^{(n)}$

نتیجه می شود که i بازدهی سمت راست اجتناب آشنا ندارد .

خط ۱۸، ۹۷ پر فیلم شد .

ویدئو ایجاد شد .

۲۴

ویدئو ایجاد شد .

ویدئو ایجاد شد .

PAPCO

پیمائش گراف:

$$G = (V, E)$$

$$\Theta(|V| + |E|)$$

DFS } الگوریتم
BFs }

Depth-First-search(G, v)

جزئی

1. DFS(G, v)

$$\rightarrow \Theta(|V| + |E|)$$

2. For $i = 1$ to $|V(G)|$ do

3. if visited[i] == false then

$$\Theta(|V|(|V| + |E|))$$

4. DFS(G, i)

$$\Theta(v_1 + E_1) + \Theta(v_1)$$

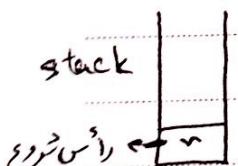
$$\Theta(v_2 + E_2)$$

$$\Theta(v_r + E_r) + \Theta(v_r)$$

$$\Theta(v_p + E_p) + \Theta(v_p)$$

زمانهای حرسی داری دارد که از دردادرد شده اطمینان بررسی کند بعد اکنون برعکس!

پیاده سازی غیر بازگشتی: DFS



unRecursive-DFS(G, v) $\rightarrow \Theta(|V| + |E|)$

1. Create empty stack S

2. $S.push(v)$

3. while $S.empty() \neq true$ do

4. $w = S.pop()$, $visited[w] = true$

5. if $visited[w] = false$ then

6. $visited[w] = true$

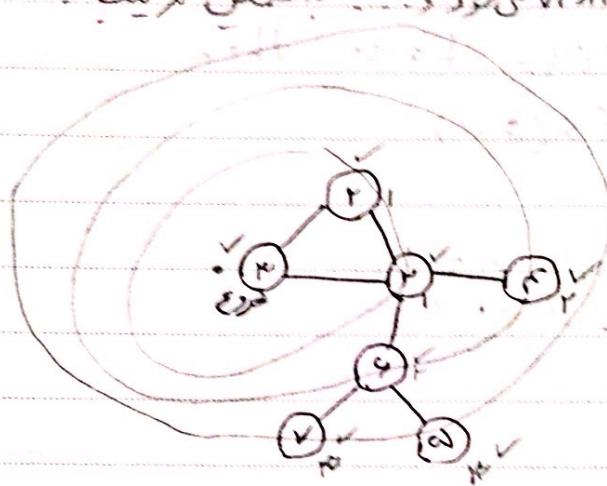
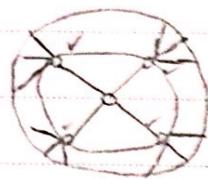
PAPCO

x0

3. for each u that is a neighbor of w do
 6. if $\text{visited}[u] == \text{false}$ then
 7. $s.\text{push}(u)$

الخطوات في الزيارة : BFS

الخطوات في الزيارة من المقدمة إلى النهاية حسب الترتيب الذي تم زيارته.



مثال :

unRecursive_BFS (G, v)

$O(|V| + |E|)$

1. Create an empty Queue Q
 2. $Q.\text{add}(v)$
 3. while $Q.\text{empty}() \neq \text{true}$ do
 4. $w = Q.\text{del}()$
 5. $\text{visited}[w] = \text{true}$
 6. for each u that is a neighbor of w do
 7. if $\text{visited}[u] == \text{false}$ then
 8. $Q.\text{add}(u)$

کاربرد BFS: کیمی پیشام در بای خودروستایی فریشی همینه و مجهودستان اد آن پیغام را به دوستان

فرموده فرمودند! ← دوچنین طور ادامه می داشت ...

مثال: زمان الگوریتم $O(n^2)$

$\text{size} \leq C \times n$ مرتباً سازی اعدام ← مساله ①
نعدادیت های عدد $\text{size} \leq M$ ضرب اعدام بیتی ← مساله ②

$$\text{size} = C \times |V| + C' \times |E|$$

خط در مرتب کردن

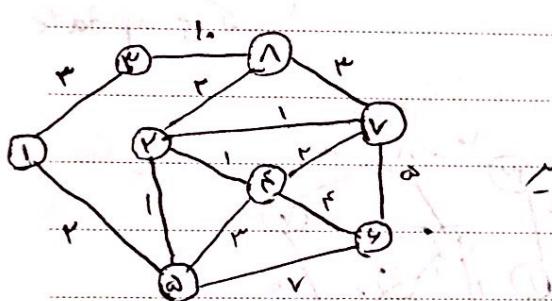
۹۷، ۸، ۲۸

حل

پیدا کردن کوتاه ترین مسیر در گراف:

مسیر ساده (نقطه تکراری ندارد)

مسیر غیر ساده (در دارد)



طول مسیر = مجموع وزن یال های مسیر

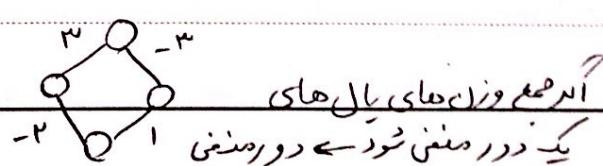
الگوریتم BFS ← کوتاه ترین مسیر فقط برای وزن یال های ۱

باید وزن های تمامی \rightarrow single source \rightarrow از یک رأس مصنوب همی کنند.

all pairs \rightarrow از هر رأس به هر رأس \rightarrow Floyd-warshall

الگوریتم

P4PCO



مجموع وزن یال های

که در منطقه شود \rightarrow در منطقه ۱

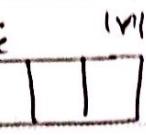
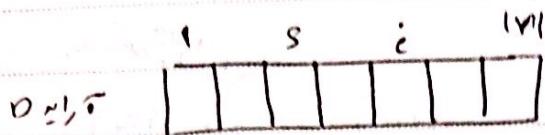
* اگر در مسیر نداشته باشیم که تا هفتمین مسیر می‌گذاشتم، مسیری نداشتم است.

۱۷۱-۳ مرندار

نهازی‌های

Dijkstra's tra

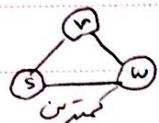
راهنمایی شروع:



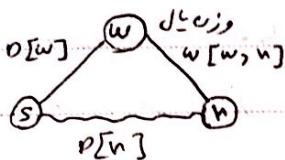
۱۷۱

$D[i]$ در نهایت طول کوتاه‌ترین مسیر از S به i خواهد بود.
مقداردهی اولیه $D[i] = \infty$

$$D[i] = \begin{cases} \infty & \text{اگر از بین ۳ مسیر نداریم} \\ w[S, i] & \text{اگر از بین ۳ مسیر ۱ مسیر داریم} \end{cases} \rightarrow O(1)$$



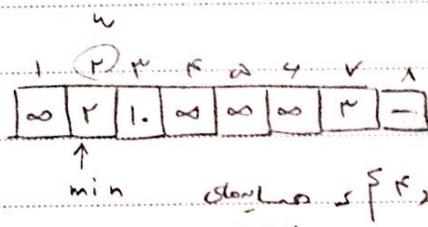
خونک نماید و رأس است که کمترین مقدار D را دارد. (D کمینه است)
طول کوتاه‌ترین مسیر از S به w همین $D[w]$ است و کوتاه‌ترین نوود.



$$D[v] = \min \left\{ D[v], D[w] + w[w, u] \right\}$$

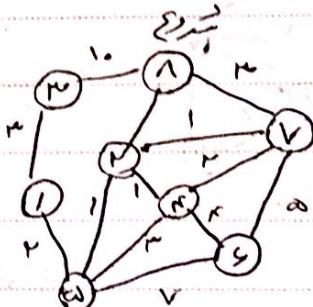
برای هر رأس v که همچنان w است، $D[v]$ را افزایش بخواهیم!

update



$\infty \ 2 \ 1. \ \infty \ \infty \ \infty \ 3 -$

\min

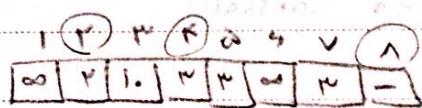


صلال

$$D[\ell] = \min \{ D[\ell], D[r] + w[r, \ell] \} = 3$$

$$D[v] = \min \{ D[v], D[r] + w[r, v] \} = 1$$

$$D[\alpha] = \min \{ D[\alpha], D[r] + w[r, \alpha] \} = 2$$



دوسته شروع $\leftarrow \min \{ \infty, 1, 2, 3 \}$

PAPCO کمینه داشتادا و لطفاً انتخاب کنیم

Q6. $D[s] = \min\{D[u], D[v] + w[u, s]\}$ $\forall u, v \in V$
 $w \in E$

$$D[u] = \min\{D[u], D[v] + w[v, u]\} \quad \forall u$$

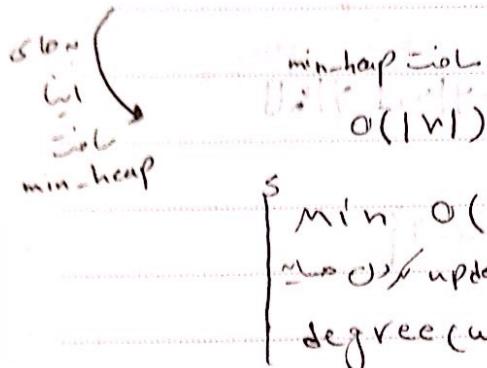
$$D[v] = \min\{D[v], D[w] + w[w, v]\} \quad \forall v$$

$$D[w] = \min\{D[w], D[x] + w[x, w]\} \quad \forall x$$

$w \in V \setminus \{s\}$ $\min \rightarrow O(|V|)$

$w \in V \setminus \{s\}$ update $\rightarrow \text{degree}(w)$

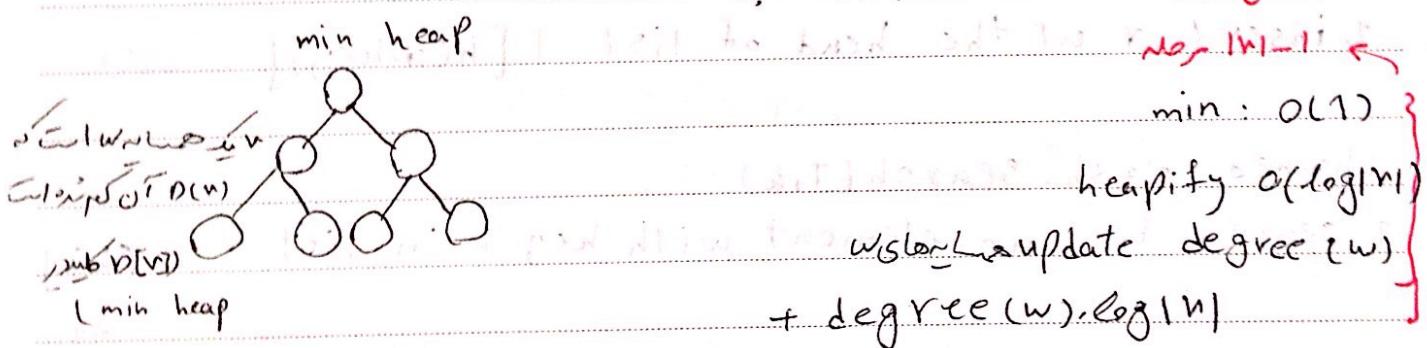
$$\text{Time} = O(|V| - 1) \times |V| + \sum \text{degree}(w) \rightarrow O(|V|^2) = O(|E|)$$



$$\text{Algorithm: } O(|E| + |V| \log |V|)$$

qV, qW

(due $O(|V|^2)$) : Dijkstra



~~PARCO~~ $O(|V| + |V| \cdot \log |V| + |E| + |E| \cdot \log |V|) = O(|E| \cdot \log |V|)$

✓

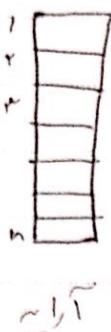
Subject:

Year:

Month:

Date:

:(چیزی کوچک جدول Hash Tables



$A[i].info$

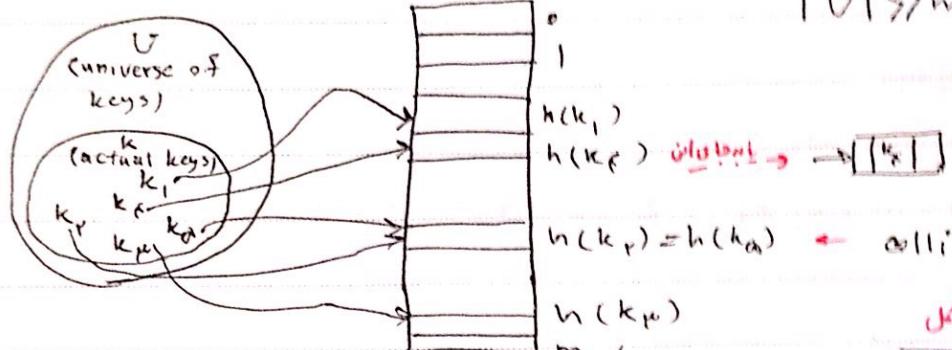
$O(1)$

و بحث

$O(1)$ SAdd
search
Del

T

$|V| \gg m$



برای این مثال

Collision

hash function

$$h: V \rightarrow \{0, 1, 2, \dots, m-1\}$$

(chaining) نکاری کردن از لیست پرینی طرفه است *
کلیک را برای رفع مکالمه

- و پی $O(1)$ این حال رساند هات متوسط $\Theta(|V|)$ این دفعه $\Theta(|k|)$

chained_Hash_Insert(T, x)

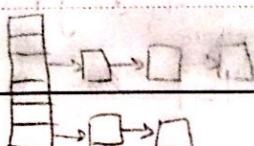
1. insert x at the head of list $T[h(x.key)]$ $O(1)$

chained_Hash_Search(T, k)

1. search for an element with key k in list $T[h(k)]$

لینک order

P4PCO chaining:



chained-Hash-delete (T, x)1. delete x from the list $T[h(x.key)]$ $\Theta(1)$ بنزرن اشتن اشارهگر به x در لیست پر نمای در طرفه در (1) انجام دارد.

تعداد عناصر

 $\frac{n}{m}$ تعداد خانه های جدول درهم \rightarrow سازیload factor $> 1 \Rightarrow$ تعداد عناصر در یک خانه بیشتر از یک است.

خانه خالی داشته باشیم

load factor $< 1 \Rightarrow$ تعداد عناصر در یک خانه کمتر از یک است.

خانه خالی را بین

 $\alpha = \frac{n}{m}$ $T[j]$: تعداد عناصر در خانه j تعداد عناصر کل $n = n_0 + n_1 + n_2 + \dots + n_{m-1}$ $E[n_j] = \alpha \approx \frac{n}{m}$ "فرض درهم سازی یکنواخت ساده"متانکن n_j (simple uniform hashing)(اصیر یافتن n_j) فرضی کنیم هر یک خانه با احتمال مساوی بجزاند در هر کدام از m خانه میتوان قدر اگربر.

: search

11.1) در یک جدول درهم سازی که دلایل $\Theta(n^2)$ برای chaining حلی جدول را که مستحبی نامرفق زنیمتوجه $\Theta(1 + \alpha)$ در با فرض درهم سازی یکنواخت ساده

$$\bar{x} = \sum_{x_i} p(x_i) x_i = x_0 + x_1 + \dots + x_m$$

$$\sum_{x_i} \frac{1}{m} \alpha = \alpha$$

حقیقت ۱۱.۲ در یک مدول در هم سازی که مقادیر chaining حلی ندارد، یک حسبتگری موفق زمان متوسط $(1+\alpha)\theta$ دارد. بافرض در هم سازی تک راهت ساره

فرضی: هر کدام از عنصر با احتمال $\frac{1}{n}$ بیواند حسبتگر شود.

$$x_{ij} = I\{h(k_i) = h(k_j)\} \text{ متغیر تصادفی}$$

$$E\left[\frac{1}{n} \sum_{i=1}^n \left(1 + \sum_{j=1}^n x_{ij}\right)\right] = \frac{1}{n} \sum_{i=1}^n \left(1 + \sum_{j=1}^n E[x_{ij}]\right)$$

$$\frac{1}{n} \sum_{i=1}^n \left(1 + \sum_{j=i+1}^n \frac{1}{m}\right) = 1 + \sum_{i=1}^n \frac{(n-i)}{m} = 1 + \frac{1}{m} \sum_{i=1}^n (n-i)$$

$$1 + \frac{n(n-1)}{m} = 1 + \frac{n-1}{m} < 1 + \frac{\alpha}{2}$$

جلسه ۵، ۹۷/۹/۲۰ آغاز شد.

۹۷/۹/۱۰.

جلسه ۶: دو شیوه ملائیک طرایق الگوریتم:

divide & conquer

(۱) تقسیم مسئله به زیرمسئلهای کوچکتر

(۲) حل بازگشتی زیرمسئلهای

(۳) ترتیب جواب زیرمسئلهای ویدست آوردن جواب مسئله اصلی

fact(n)

مسئل: فاکتوریل:

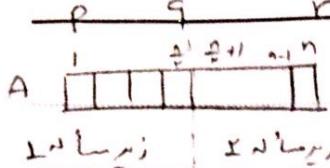
```
if n==1 then return 1
return fact(n-1)*n
```

?

PAPCO

Subject

Date



مثال: مجموع اعداد از p تا r : $A[p:r] = \sum_{i=p}^r A[i]$

$$q = \left\lfloor \frac{p+r}{2} \right\rfloor$$

$T(n) = O(1)$

$\text{Max}(A, p, r)$

1. if $p=r$ then return $A[p]$

2. if $p>r$ then return error

$$3. q = \left\lfloor \frac{p+r}{2} \right\rfloor$$

$$4. \text{max1} = \text{Max}(A, p, q)$$

$$5. \text{max2} = \text{Max}(A, q+1, r)$$

6. if $\text{max1} > \text{max2}$ then return max1

7. else return max2

$$T(n) = T\left(\frac{n}{2}\right) + O(1) \rightarrow O(n)$$

spouse

$\text{Max}(A, p, r)$

1. if $p=r$ then return $A[p]$

2. if $p>r$ then return error

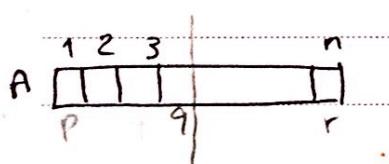
$$3. \text{max} = \text{Max}(A, p, r-1)$$

4. if $\text{max} > A[r]$ then return max

5. else return $A[r]$

$$T(n) = T(n-1) + O(1) = O(n)$$

مساله مرتب سازی: مرتب سازی مسیری



$\text{merge_sort}(A, p, r)$

1. if $p=r$ then return

PqPCO

rg

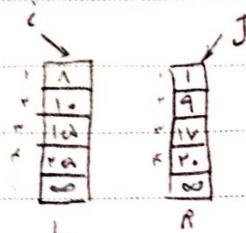
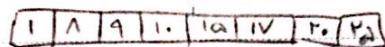
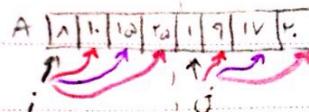
2. $q = \left\lfloor \frac{p+r}{2} \right\rfloor$

3. merge-sort(A, p, q)

4. merge-sort($A, q+1, r$)

5. merge(A, p, q, r)

sort كل دورة مرتاح مرتاح



merge(A, p, q, r)

1. $n_1 \leftarrow q-p+1, n_2 \leftarrow r-q$

2. for $i \leftarrow 1$ to n_1 do

3. $L[i] \leftarrow A[p+i-1]$

4. $L[n_1+1] \leftarrow \infty$

5. for $i \leftarrow 1$ to n_2 do

6. $R[i] \leftarrow A[q+i]$

7. $R[n_2+1] \leftarrow \infty$

8. $i \leftarrow 1, j \leftarrow 1$

9. for $k \leftarrow p$ to r do

10. if $L[i] < R[j]$ then

11. $A[k] \leftarrow L[i]$

12. $i++$

13. else $A[k] \leftarrow R[j]$

14. $j++$

Subject

Date

p1	10	11	12	13	14	15
A	10	11	12	13	14	15

طریقه از فقط چند از آن نوشته شده است.

بعنده همین مرتب است.

p1	10	11	12	13	14	15
A	10	11	12	13	14	15

p1	10	11	12	13	14	15
A	10	11	12	13	14	15

p1	10	11	12	13	14	15
A	10	11	12	13	14	15

p1	10	11	12	13	14	15
A	10	11	12	13	14	15

$O(n) = \text{merge}$ زمان *

9V, 9, 1V

A	i	j	r

نمایش

عمل بارگذاری $M(\frac{n}{f})$ +

merge sort (زمان) $O(n)$ +

merge sort کل $T(n) = 2T(\frac{n}{f}) + O(n) = O(n \lg n)$

برای کل کاربرد

$\Theta(n \lg n)$ زمان merge sort را داشت.

: Insertion-Sort

A	i	j	r

حالت دوم (نهایی میکاری)

این الگوریتم insertion sort دیگر بارگذاری نمایش

Insertion-Sort(A, p, r)

1. if $p >= r$ then return

2. Insertion-Sort(A, p, r-1)

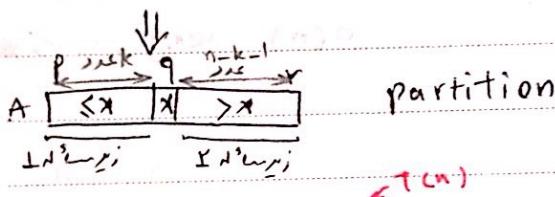
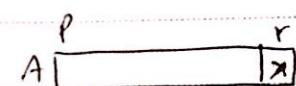
برای کل بارگذاری
نمایش

۴

3. $\text{key} \leftarrow A[r] , i \leftarrow r-1$
 4. while $i >= p$ and $\text{key} < A[i]$ do مترین حالت : $O(n)$
 5. | $A[i+1] \leftarrow A[i]$ مترین حالت : $O(n^r)$
 6. | $i--$ حالت متوسط : $O(n^r)$
 7. $A[i+1] = \text{key}$

: quick sort

زایر رایم قسم تقسیم کنیم



Quick-sort(A, p, r)

1. if $p >= r$ then return $O(1)$

2. $q \leftarrow \text{partition}(A, p, r)$ $O(n)$

3. Quick-sort($A, p, q-1$) $T(k)$

4. Quick-sort($A, q+1, r$) $T(n-k-1)$

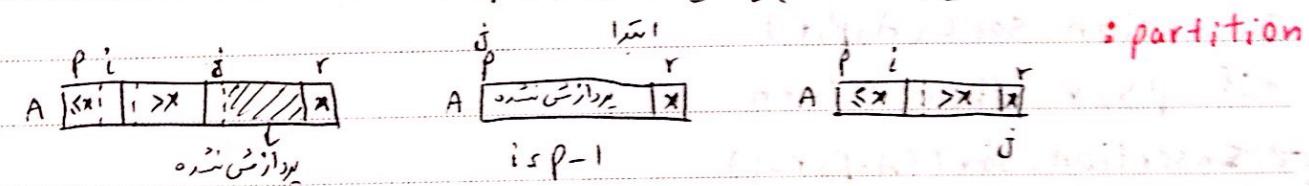
چون قسم سنتی بگیریم از قسم سنتی برآید است و \rightarrow نهایی ۵. بگردید
که توان هر ۲ قسم را sort کرده و در آنارهم گذاشت (نهایی به مرتبه ۸ و ۷ ترتیب شود).

نحوه:

* بزرگ ترین قسم کوچک دیگر با هم ممکن نیست

* در اصل اوری گام تقسیم است و بیشترین زمان رایج بود، در و در اصل

گردید گام ترکیب اسکوپیتی زمان رایج بود.



PCPQ

partition(A, p, r)

1. $i \leftarrow p, i \leftarrow p+1, x = A[r]$

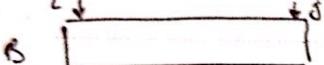
2. for $j \leftarrow p$ to $r-1$ do

3. if $A[j] \leq x$ then $\cancel{A[i:j]} = O(n)$

4. $i \leftarrow i + 1$

5. swap($A[i], A[j]$)

6. swap($A[i+1], A[r]$)



دستورات دیگر را در اینجا بخواهید!

: Quick Sort

$$T(n) = T(k) + T(n-k-1) + O(n)$$

محبّت روی صدای k :

$$0 \leq k \leq n-1$$

$$O(n) \rightarrow C + O(n)$$

فرمی: $T(0) = C$

$$T(n) = T(0) + T(n-1) + O(n)$$

عنوان آرایه (\dots)

مرتب شده نظری

اسید

فرمی

$$T(n) = T(n-1) + T(0) + O(n) = O(n)$$

آرایه در دردی

مرتب شده

مقداری است

فرمی

$$T(n) = T\left(\frac{n}{p}\right) + T\left(\frac{n}{p}-1\right) + O(n) = pT\left(\frac{n}{p}\right) + O(n) = O(n \lg n)$$

زمان در حالت min

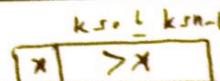
$$T\left(\frac{n}{p}\right) + T\left(\frac{n}{p}-1\right)$$

و سطوحی رصد.

مقدار ابرابر است.

Subject

Date



$O(n \lg n)$ متران partition

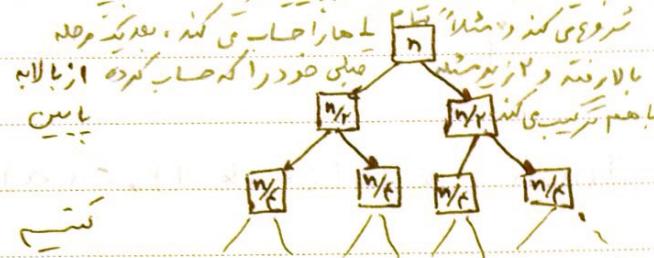
$O(n^2)$ غیر متران partition

9V, 9, 1V

Dynamic programming (روشی برتر سه روشی بروی):

divide & conquer

نه شروعی کند و تا آن زمین عصرست
یعنی رسید بعد درباره مرغله بزمحل
بالای آید (ادل سیم بعد تر لب)
اما مکن است یک ریز مسئله قبلاً
مرغی حل شده باشد و لی چون خاص مطه
از یابین، بذار در دوباره آن را حل کن



ضخیاز
پاسی:

بالا

یکمیش DFS ترتیب

متادرات های Dynamic divide

1) در Dynamic اطلاعات را ذخیره کن.

2) در Dynamic فقط از یابین به بالای رود

* مسئلهای که با حلی کنیم بعده دارای دیگر "زیر ساختار چینه" (optimal substructure) هستند.

چنان یکنینه از هواب بگینزیر مسئله

از D.P برای حل مسائل بحثیه سازی استفاده کنیم.

هواب غیر یکنینه نیست.

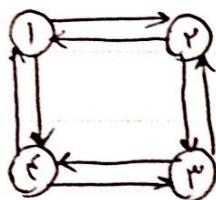
مثال از یک مسئله که در دیگر زیر ساختار چینه دارد:
ساده کرکارهای مسیر در گراف



مسیر کل کوتاهترین مسیر است
پس هر کدام از آن هم کوتاهترین است.

PAPCO

مثال از که می‌دانم که در گروه زیر ساخت رجیسٹر را دارد:



: طولانی‌ترین مسیر به ۲

اگر از اینجا بد اینم طولانی‌ترین →
مسیر تا به ۲ نگهاد نیست و ..

نحوی آن

مسئله‌ای که با 0.9 بعنوان مثالی فراهم مل نیم:

مسئله پرانتز گذاری بحینه در ضرب n ماتریس:

$$((M_1 \times M_2 \times M_3) \dots \times M_{n-1} \times M_n)$$

$$M_1 \times M_2 \times M_3 \times M_4 \\ 1 \times 2 \quad 2 \times 3 \quad 3 \times 1 \quad 1 \times 2$$

$$P(n) = \sum_{k=1}^{n-1} P(k) P(n-k) \\ \text{عدد علاوه} = \frac{1}{n+1} (2^n)$$

راه را

$$(M_1 \times M_2)(M_3 \times M_4)$$

$$1 \times 3.$$

$$\diagdown \quad \diagup \quad \text{ضریب ۴۰۰.}$$

$$1 \times 2 \times 2 \times 1 \quad 1 \times 2 \times 2 \times 1 \quad 1 \times 2 \times 2 \times 1$$

$$M_1 \quad M_2 \quad M_3 \quad M_4 \quad \text{ضریب ۴۰۰.}$$

$$1 \times 2 \quad 2 \times 1 \quad 1 \times 2 \times 1 \quad 1 \times 2 \times 1$$

$$P \begin{array}{|c|} \hline q \\ \hline \end{array} \times \begin{array}{|c|} \hline r \\ \hline \end{array} = P \begin{array}{|c|} \hline q \cdot r \\ \hline \end{array}$$

$$M_{p \times q} \times M_{q \times r} = M_{p \times r} \quad \text{تکرار ضرب} \quad P \cdot q \cdot r$$

$$(M_1 \times ((M_2 \times M_3) M_4))$$

$$1 \times 2 \quad 2 \times 3 \quad 3 \times 1 \quad 1 \times 2 \quad \text{ضریب ۱۰۰.} \quad \text{ضریب ۲۰۰.} \quad \text{ضریب ۱۰۰.}$$

$$M_1 \quad M_2 \quad M_3 \quad M_4 \quad 1 \times 2 \quad 2 \times 3 \quad 3 \times 1 \quad 1 \times 2 \quad \text{ضریب ۱۰۰.} \quad \text{ضریب ۲۰۰.} \quad \text{ضریب ۱۰۰.}$$

دارد optimal substructure

$$P_{i,j} = \underset{d_{i-1} \times d_k}{\text{صریح}} \times \underset{d_{k+1} \times d_j}{\text{صریح}} ((M_i \times M_{i+1} \times \dots \times M_k) \times (M_{k+1} \times \dots \times M_j))$$

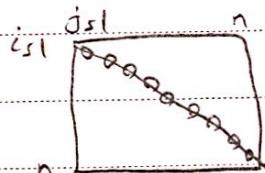
نکار هندسی های کل در ماتریس

$$P_{i,j} = \underset{(P_{i,k} \times P_{k+1,j})}{\text{صریح}} + \underset{(P_{i,k} \times d_k \times d_j)}{\text{صریح}} + \underset{(P_{k+1,j} \times d_k \times d_j)}{\text{صریح}}$$

$$C_{i,j} = \min \{ C_{i,k} + C_{k+1,j} + d_{i-1} \times d_k \times d_j \}$$

$$i \leq k \leq j$$

کوچکترین زیرمسئله $P_{i,j} = C_{i,j}$



نهایی در ماتریس

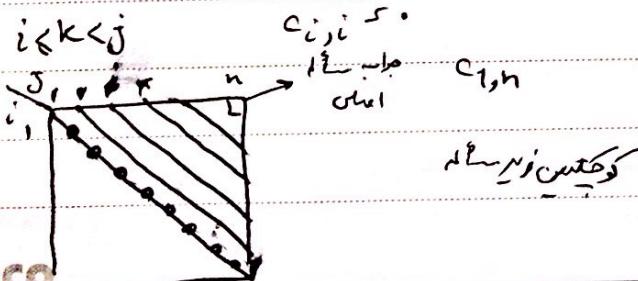
طبقه بندی

۹V, ۹, ۱۹

ادامه مسئله پرانتز لذاری مربوط ماتریس ها:

$$\text{از } P_{i,j} : \frac{M_i \times M_{i+1} \times \dots \times M_k \times M_{k+1} \times \dots \times M_j}{d_{i-1} \times d_j} \times d_{j-1} \times d_j$$

$$C_{i,j} = \min \{ C_{i,k} + C_{k+1,j} + d_{i-1} \cdot d_k \cdot d_j \}$$



$$M_1 \times M_r \times M_p \times M_f$$

d. $d_1 d_r d_p d_f$

		j		1	r	p	f
		1	r	1	r	1	r
1	0	1000	1000	1000	1000	1000	1000
r	0	1000	1000	1000	1000	1000	1000
p		0	1000	1000	1000	1000	1000
f			0				

$$\frac{n(n+1)}{r} \leq \frac{n+r}{r}$$

مقدار k را از زیر

$$C_{1,r} = C_{1,1} + C_{r,r} + d_1 d_r d_p = 0 + 0 + 0 \times 10 \times 10 = 100$$

$$C_{r,p} = C_{r,r} + C_{r,p} + d_1 d_r d_p = 0 + 0 + 0 \times 10 \times 10 = 100$$

$r \leq k \leq r$

$$C_{r,f} = C_{r,r} + C_{r,f} + d_r d_p d_f = 0 + 0 + 0 \times 10 \times 10 = 100$$

$r \leq k \leq f$

$$M_1 \times M_r \times M_p$$

$$C_{1,r} = \min_{1 \leq k \leq r} \{ C_{1,k} + C_{k+1,r} + d_1 d_k d_p \}$$

$$M_1 \times (M_r \times M_p) \leftarrow \begin{cases} k=1 & C_{1,1} + C_{r,r} + d_1 d_r d_p = 0 + 0 + 0 \times 10 \times 10 = 100 \\ \dots & \dots \\ k=r & C_{1,r} + C_{r,r} + d_1 d_r d_p = 100 + 0 + 0 \times 10 \times 10 = 100 \end{cases}$$

$(M_1 \times M_p) \times M_r \leftarrow$

$$M_r \times M_p \times M_f$$

$$C_{r,f} = \min_{r \leq k \leq f} \{ C_{r,k} + C_{k+1,f} + d_1 d_k d_f \}$$

$$M_r (M_p M_f) \leftarrow \begin{cases} k=r & C_{r,r} + C_{r,p} + d_1 d_r d_f = 0 + 0 + 0 \times 10 \times 10 = 100 \\ \dots & \dots \\ k=f & C_{r,f} + C_{f,p} + d_1 d_f d_f = 100 + 0 + 0 \times 10 \times 10 = 100 \end{cases}$$

پرسو

۴۴

Subject

Date

$$M_1 \times M_r \times M_p \times M_s \quad M_1 \times ((M_r \times M_p) \times M_s)$$
$$C_{1,r} = \min_{1 \leq k < r} \{ C_{1,k} + C_{k+1,r} + d_r \cdot d_k \cdot d_s \}$$

$$M_1 \times (M_r \cdot M_p \cdot M_s) \leftarrow k+1 \quad C_{1,1} + C_{r,s} + d_r \cdot d_p \cdot d_s + 1.00 + 1.00 \times 1.00 = 5.00$$

$$= \min_{(M_r \times M_p) \times (M_s \times M_s)} \{ C_{1,r} + C_{r,s} + d_r \cdot d_p \cdot d_s = 1.00 + 1.00 + 1.00 \times 1.00 = 4.00$$

$$(M_1 \cdot M_r \cdot M_p) \times M_s \leftarrow k+1 \quad C_{1,p} + C_{s,s} + d_r \cdot d_p \cdot d_s = 1.00 + 1.00 \times 1.00 = 3.00$$

D.P جواب مسأله خوارزمی مسائل

$$= O(n^3) \times O(n) = O(n^4)$$

$$1 \times n + 1 \times (n-1) + 1 \times (n-2) + \dots + (n-1) \times 1$$

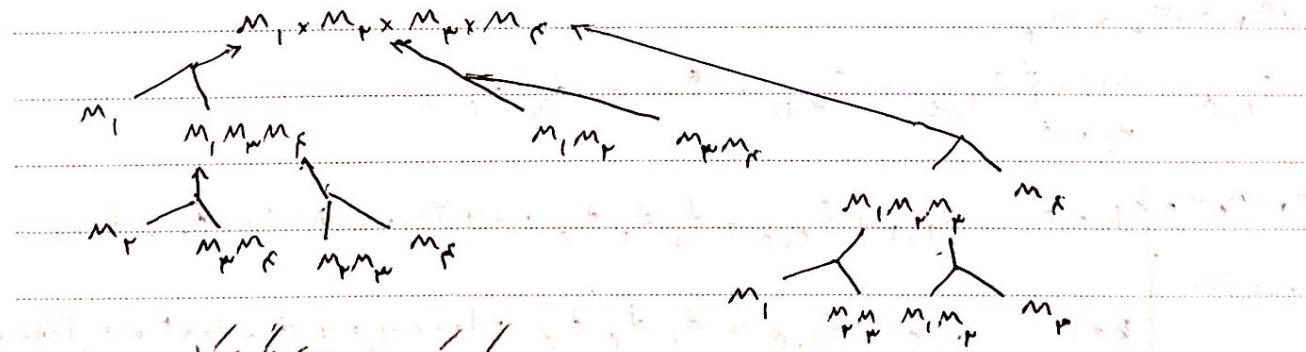
$$n + (n-1) + (n-2) + \dots + 1 = \frac{n(n+1)}{2}$$

$$(n-1) + (n-2) + \dots + 1 = \frac{(n-1)(n-2)}{2}$$

$$(n-1) + (n-2) + \dots + 1 = \frac{(n-1)(n-2)}{2}$$

: (divide and...) مراد

$$T(n) = \sum_{k=1}^{n-1} (T(k) + T(n-k)) = O(n^2)$$



D.P خوب است اما غلط کار می شود و لزومی این خوارزمی بجهت این رسمیت آن کاربری برداشته شود

PAPCO

مراد

روش گریدنی (Greedy):

$$M_1 \times M_2 \times M_3 \times M_4$$

greedy choice

$$c_{i,j} = \min_{1 \leq k \leq j} f_{i,k}$$

مثل دلخواهی موند راهی ما فرمیان

کمین را انتخاب کرده و فرمی کنیم

حینه است و آن را هم کنیم به باید است (کمین) که راهی کمین برده باشد (علان است
فرمی ما است باشد).

۹۷/۹/۲۸

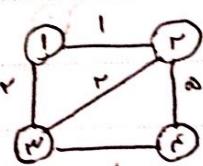
مسأله کرتا هستین مسین

Floyd-warshall : all pairs

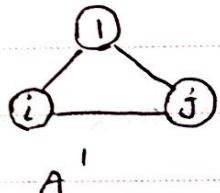
(مبتنی به زوپس برنامه ریزی پرسا)

اگر صدی هزار دهی کناریم

	1	2	3	4
1	0	1	2	∞
2	1	0	2	2
3	2	2	0	1
4	∞	0	1	0



A' ماتریس مسافت



$$A'[i,j] = \min \{ A[i,j], A[i,1] + A[1,j] \}$$

$k=1$

	1	2	3	4
1	0	1	2	∞
2	1	0	2	2
3	2	2	0	1
4	∞	0	1	0

$$A'[r,r] = \min \{ A[r,r], A[r,1] + A[1,r] \}$$

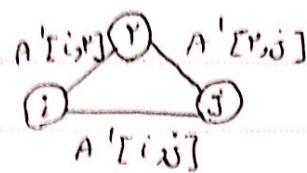
$$= \min \{ r, 1+r \} = r$$

PAPCO

۲۵

Subject _____
Date _____

$k \leq r$ A^r



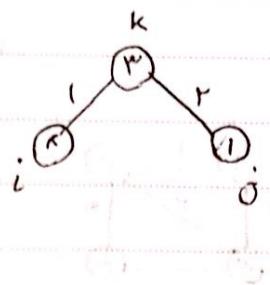
1	1	1	1	1
2	1	0	1	0
3	1	1	0	1
4	0	0	1	0

$k < r$ A^r

1	1	1	1
2	1	0	1
3	1	1	0
4	0	1	0

$k > r$ A^r

1	1	1	1
2	1	0	1
3	1	1	0
4	0	1	0



update $O(1)$ \rightarrow $O(n^2) \times O(n^2) = O(n^4)$

* $n \times n \times n \times n = O(n^4) = O(|V|^4)$

* $O(|E| \log |V|) \times |V|$

single-source

all pairs



روش خردمند
(Greedy)

مثال: ساله هر زارون اگناس بقدری میزد

R4PCO

هرفت: هر کارون با کمترین تعداد سکمه

انتخاب هریمانه: اولویت با کمترین بزرگی است.

$$s_1 > s_2 > s_3 > \dots > s_n$$

اکتس ۵۰ ترمی

۱۳۰ کم

جواب هریمانه: $1\alpha = 3 \times \alpha + 0$

کم

اکتس ۶۰ ترمی

۱۲۰ کم

جواب هریمانه: $5 \times \alpha_0 + 10 \times 1$

کم

۶۰ کم

کم

مثال نصفن هم برای نهادن جواب هریمانه این است که در همین مثال کمترین زمان را باشیم و بسیار بخوبی بگوییم.

مثال: اگر (T_i) روی یک ماشین کمترین زمانی که برازنه اصرایی Task باشد شامل شورکابران

Task بعدی را اجرا کرد.

Task i: t_i : زمان پروازی

i: ترتیب اصرایی مختلف

$$\langle T_1, T_2, T_3 \rangle$$

T_1 : زمان پاسخ

$$T_2 = t_1 + t_2$$

صلانی زمان

$$+ t_2 + t_3$$

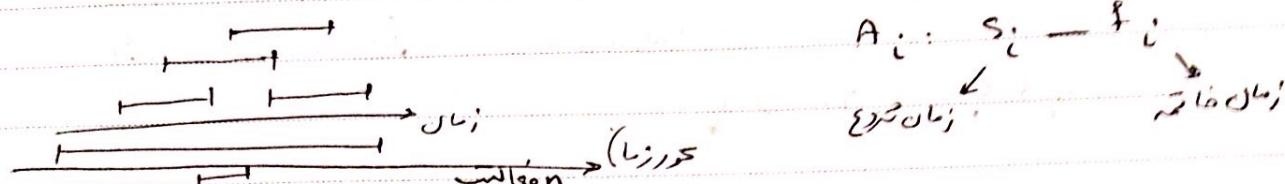
$$T_3 = t_1 + t_2 + t_3$$

mins?

انتخاب هریمانه: اصرایی Task ها به ترتیب صدردی زمان برازش

* حسین است

مثال، انتخاب فعالیت



۹۷, ۹, ۲۴

$$A_i = (s_i, f_i)$$

نحوه انتخاب \rightarrow
لحظه شروع

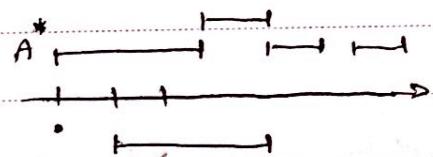
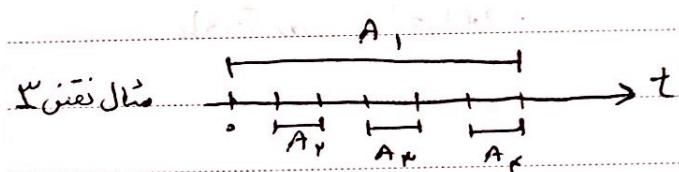
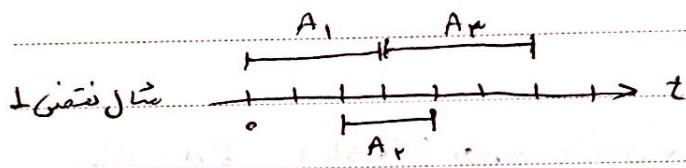
هدف: انتخاب بیشترین فعالیت که باهم همپوشان نباشند.

انتخاب حریصانه ①: اولویت انتخاب با فعالیت کوتاهتر

انتخاب حریصانه ②: اولویت انتخاب با فعالیت است که کمترین همپوشان باقیگاردارد.

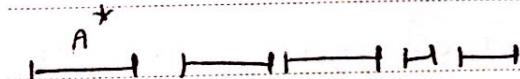
انتخاب حریصانه ③: اولویت انتخاب با فعالیت که بیکمتر دارد.

انتخاب حریصانه ④: اولویت انتخاب با فعالیت که بیکمتر دارد.



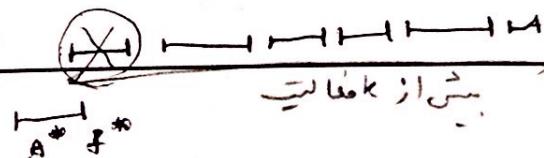
$$A_i \quad f^* < f_i, \quad A^* = (s^*, f^*)$$

فرض: تعداد فعالیت بینه بافرض انتخاب A^*

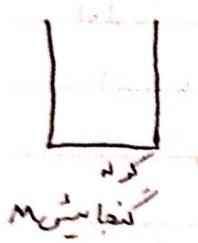


ادعا: هیچ جواب دیگری که بیش از کافعالیت داشته باشد $\Rightarrow A^*$ در آن نباشد و صریح باشد.

: اثبات ادعای پرونده



مثال: مسأله کوله سک (knap sack)



مجموع ایجاد.

$$O_i = (w_i, v_i)$$

نخاست

ارزش تر، وزن دارد.

$$w_i \leq M$$

برای ایجاد
دافت روی

$$\max \sum v_i$$

برای ایجاد
دافت روی

دافت روی

که کسی یا کامل را داشت یا بیرون نداشته باشد.

$\in NP\text{-hard}$

می تواند سری از ترکیبیات داشته باشد که در آن کسر است.

$\in P$

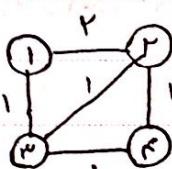
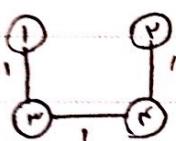
$$\frac{v_i}{w_i} \geq \text{جایزه}$$

مسأله ۲ ساخته دارد!

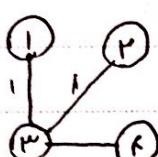
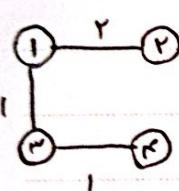
MST

مثال: مسأله پیکاردن درخت بروشان کمین (minimum spanning Tree)

$$\sum w = 3$$

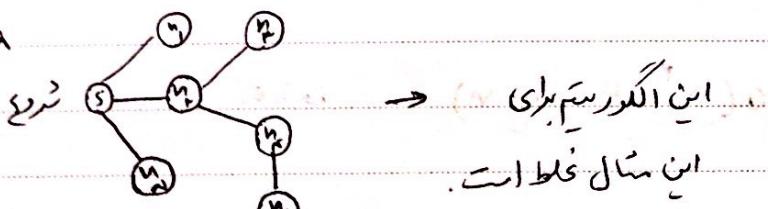


$$\sum w_i = 4$$



$$\sum w_i = 3$$

Dijkstra



این الگوریتم برای

این مثال خطا است.

چنانچه رئیس های باید این

P4PCO

مسئله این است که چه یال های را انتخاب کنیم.

$$\begin{matrix} P_{0,0} \\ P_{0,1} \\ P_{1,0} \\ P_{1,1} \end{matrix}$$

$$\begin{matrix} R_{0,0} \\ R_{0,1} \\ R_{1,0} \\ R_{1,1} \\ P_{0,0} \end{matrix}$$

اولویت با انتخاب یال کینه است.

۱- $n-1$ یال و محدوده ارد.

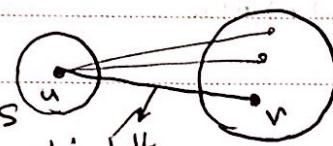
لئے اگر $n-1$ انتخاب کنیم و دور سلسل شود، همینه می شود و این هواب مندام است.
اولویت با انتخاب یال کینه است. اگر انتخاب یال پایا مجاہین که قبل از انتخاب شده سلسل دور
می دهد این یال را انتخاب نمی کنیم **Kruskal algorithm**

اینها بزرگتر

AVVAH

ادامه صریحانه:

سایه درخت پوشای کینه (MST) **prim**
که سر رأس را از آرتوی دویتی کن که هر رأس صابرد تویی در
T خالی شود.



قضیه یال کینهین در حوزه MST ؟

MST حوزه

: **Kruskal algorithm**

sort یالها ← انتخاب یال و ساختن جمله ← **Kruskal**

باید

$$E \log |E| = E \cdot \log n$$

update $\alpha(n)$

$O(n)$ DFS $\times |E|$

DFS راهارم (۱) $O(E, n) = O(n^2)$

(۲) راهدار $O(n^2 + E \cdot \log n)$ label

PAPCO

set operations:

E. O(log n)

Find(S, x) اگر درخت مستان باشد $\rightarrow O(\log |S|)$

union(S, S_r) $O(1)$

add(S, x) $O(1)$

Del(S, x)

prim: $O(n \cdot E)$

جایگزینه ۷۰ برآورده افتتاحی نیست.

بیکاردن min در لیست L (لیست با احتمالات توزیعی)

$|E|$

$\sum_{w \in V} \text{degree}(w)$ مشخص افتتاحی بودن یا نبودن طالعای n

$$L = O(|E|)$$

برای همین درون زمان:

اگر L در تکراری شود، زمان: $\min \text{ heap}$ \rightarrow heap

$$\sum_{w \in V} (\text{degree}(w) \cdot \log |E|) + |E| \times (O(1) + O(\log |E|)) =$$

زمان صرف را نمایم

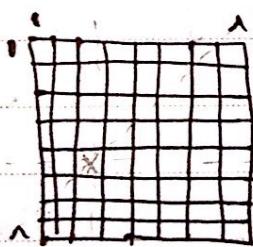
$$|E| \cdot \log |E| + |E| \cdot \log |E| = E \cdot \log n + n \cdot \log n = E \cdot \log n$$

دش جستجوی فناوری حالات:

مثال: سکله آوزیر:

وزیر را مطهری قدرار دهد که هیچ وزیری دیگری را خدیده نباشد.

که وزیر سه سطه و سه کوکو و خطر را خدیده کند.



64 حالات \rightarrow وزیر اول

در حالات کلی بود در نظر
گرفتن غیرقابل پیوند

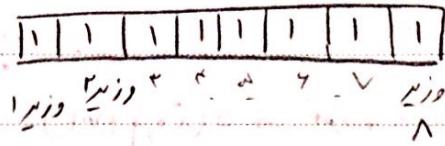
64 حالات \rightarrow وزیر دوم

کل حالات = 64^2

$$(n \times n)^n = n^n$$

PAPCO

سازنده : رامین

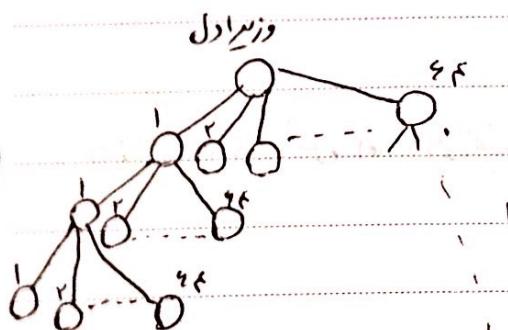


۱۶

و خارجی راههای حالات
لماکی کنیم.

۹۷، ۱۰، ۳

درس خسته‌بزی فضای حالات:



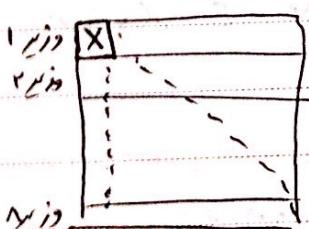
back tracking:

مانند وزیر

پیمائش DFS

+ وقت از بیش تا بگ را می‌کنیم یعنی ۸ وزیر انتخاب شده است.
حال می‌کنیم کنیم الگوریتم بود که level به عقب برگرد و انتخاب
کننده در راسته بازگر درست نشود که level به عقب
گردد و آن انتخاب را تغییر دهیم و به همین ترتیب
این کار را ادامه می‌دهیم تا ب صواب درست برسیم.
+ آن فضای حالات را کاهش دهیم، زمان الگوریتم نیز کاهش می‌یابد. ← حالاتی که امکان ندارد را حذف
کنیم.

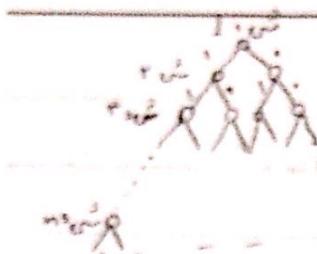
+ هر سطح فقط یک وزیری کو انداشته باشد ← هر وزیر ۸ انتخاب دارد.
وقتی وزیری انتخاب می‌شود کل سترن که در آن صراردراره حذف می‌شود، قطری که در آن قرار
دارد نیز حذف می‌شود.



+ اگر وزیر ۳ انتخاب نداشت، انتخاب ۱-۲ را عرض می‌کنیم تا درست شود.

(مثال، کربلا)

+ محبتین ساخته اند راده بدای ذخیره آن یک ماتریس ۳ بعدی است که باین
صفر است عمل کنند که مثلاً اگر صحیح وزیری در آن بنایش و تحریکی هم نباشد، صفر باشد و اگر وزیری
در آن باشد مثبت باشد و اگر تحریک روی آن باشد، منفی باشد.



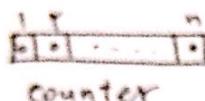
مقایی حالت: x^n

$$\frac{w_i}{w_j}$$

fractional

ساده کرده است:

کسر



حیثیت انتخاب شده

for i=0 to $x^n - 1$

$$\begin{aligned} w_i \leq M &\rightarrow \\ \text{باید} &\rightarrow \\ z_{w_i} &\rightarrow \end{aligned}$$

+ توانان و دگران، رای بمال داران!
روش branch & bound (شاخص حد):

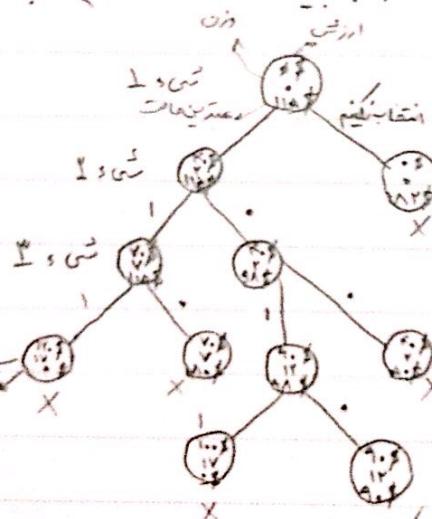
برابر	کسر	ازرس	وزن	ازرس	وزن	شیوه
$\frac{1}{2}$	۰	۴۰	۲	۱۰	۵	شیوه ۱
$\frac{1}{2}$	۱	۳۰	۵	۱۰	۱۰	شیوه ۲
$\frac{1}{2}$	۰	۵۰	۱۰	۱۰	۱۰	شیوه ۳
$\frac{1}{2}$	۱	۱۰	۵	۱۰	۱۰	گنجایش کود

از این رسم صریعانه جزوی استاد

عد بالای برای مراحلی ما \rightarrow حالت ایده‌آل \rightarrow $\frac{1}{2} \times 11 \approx 5.5$ صریعانه

جزاب چینه \leftarrow هزاب چینه و ا

این طرز حل کنیم.
ت ممکن نباشد این روش برای مسیر ۵ هزار
شیوه است را راهنمایی نهیم



کدام شاخه هر قیمت شود؟

آنچه باشد بکاهیت بکاهیت آن از شاید دیگر

کوچکتر باشد

(۱) آنچه باشد که وزن آنها بزرگتر از گنجایش کوله

است.

Subject

Date

راه جاگزینی این راه DFS روش کل شامه هاست که زمان آن نماین است (زمان بیشتری طولی کشیده) + در بدترین حالت 2^n حالت است ولی در عمل این طریق نشود.

P4PCO