

مراجع دليل:

ساختار داده:

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

② درس قسمی → نویسنده داده و الگوریتم ها

۵ انتشار بین المللی

۱۲ بیانیه

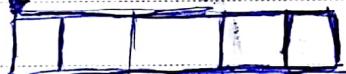
۳ ترجمه کرد

الگوریتم ۲

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



A درست



کلیل نظر الگوریتم ۲

1 $x = a$

خط = عبارت دستورات

2 for i:=0 to n do:

حذف شود

3 $x = x + r$

b c

$$T(n) = a \cdot n + b$$

lineir \Rightarrow تابع درجی از n بخطی

$T(n) = n + r$

$T(n) = n + 1$

PAPCO

$$T(n) = 1 \cdot C_1 + (n+r) \cdot C_r + C_r \cdot (n+1)$$

Subject
Date

insertion sort : ترتیب سازی درج

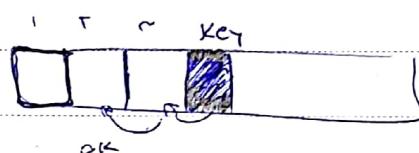
دوری: آرایه از عدد حقیقی

خرچی: آرایه A (زیر) n عدد چیزی بهوت ترتیب شده صوری (اسناد) از روشن (فریض)

Incremental

insertion sort = 1. for $i \leftarrow r$ to n do \rightarrow

2. $\text{key} \leftarrow A[j]$



$$3. \quad i \leftarrow j - 1$$

E. ~~key~~ $\leq A[i]$ and $i > 0$ [d] [f]

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

9 $i \leftarrow i-1$ }

v. $A[i+1]$ key

Engg Clrs

	n	C_n
i	$n-1$	C_i
r	$n-1$	C_r
s	$n-1$	C_s
e	$\sum_{j=r}^n (t_j + 1)$	C_e
d	"	C_d
g	$\sum_{j=r}^n t_j$	C_g
v	$n-1$	C_v

فلا ينتهي while $\text{while } \text{break} = t_j$

PAPCO (for) پاپکو

Subject

Date

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

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

$$\therefore t_j < j-1$$

$$T_n = C_1 n + C_r (n-1) + C_c (n-1) + C_e (n-1) + C_v (n-1) \quad (t_j = j-1) \quad \text{worst case}$$
$$= an + b \quad \text{میں کوئی سادہ تریں}$$

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

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

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

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

$$t_j = \frac{j-1}{r} \quad : (\text{Average Case}) \quad \text{میں کوئی سادہ تریں}$$

$$T_n = an + bn + c$$

Bubble sort

بیبل سارٹ

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 ($A[j], A[j+1]$)

}

$$T(n) \leq C_1 \cdot n + C_r \sum_{i=1}^{n-1} (i+1) + C_w \sum_{i=1}^{n-1} i$$

$$= C_1 \cdot n + \frac{n^2 + (n-1)}{2} C_r + C_w \frac{(n-1)n}{2}$$

$$= an^2 + bn + c$$

$$\begin{aligned} & \text{---} & & \text{---} \\ & 1 & n & C_1 \\ & r & \sum_{i=1}^{n-1} i & C_r \\ & \{ & \sum_{i=1}^{n-1} i & C_w \end{aligned}$$

الخطوة i تتم في $n-i$ مقارنة. كل مقارنة تأخذ C_w مدة.

مقدار المقارنات هو $\sum_{i=1}^{n-1} i$.

$\omega, o, \theta, \Omega, O$ مترادفون

little omega

little theta

teta

big omega

big O

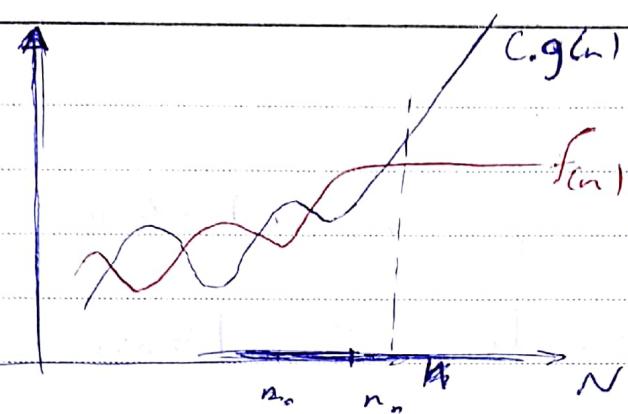
$f(n) \in O(g(n)) \Leftrightarrow \exists c > 0, \exists n_0 > 0 \mid \forall n > n_0 \quad f(n) \leq C \cdot g(n)$

$g(n)$ يمثل عبء الحساب في خطوة i .

PAPCO

Subject

Date



(C.g(n)) > f(n)

(C.g(n))

$$\text{def: } \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \infty \quad \text{if } \forall \epsilon > 0, \exists N \in \mathbb{N} \text{ such that } \forall n > N, \frac{f(n)}{g(n)} > \epsilon$$

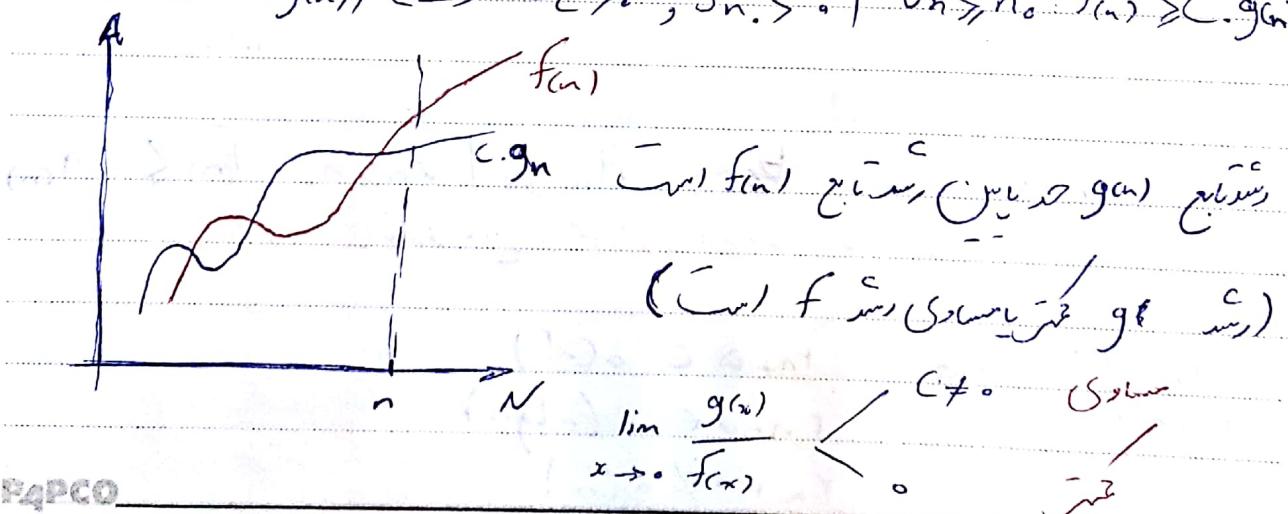
$$\lim_{n \rightarrow \infty} \frac{r_n}{n} = \infty \quad n \in O(r_n) \quad n \notin O(\sqrt{n})$$

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

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

big omega: $\Omega(g)$

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



$$\lim_{x \rightarrow \infty} \frac{g(x)}{f(x)} = 0 \quad \text{if } \forall \epsilon > 0, \exists N \in \mathbb{N} \text{ such that } \forall x > N, \frac{g(x)}{f(x)} < \epsilon$$

R&PCO

$$\textcircled{2} \quad n \in \Omega(n), \lim_{n \rightarrow \infty} \frac{r_n}{n} = r \quad \lim_{n \rightarrow \infty} \frac{r_n}{\sqrt{n}} = 0 \quad n \in \Omega(\sqrt{n})$$

Subject
Date

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

(جیسے جیسے) : teta θ

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

$$C_1 \cdot g(n) \leq f(n) \leq C_2 \cdot g(n)$$

$$n \in \theta(r_n), r_n \in \theta(n), n \in \theta(\sqrt{n}), n \in \theta(n^x)$$

(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

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

که دلیل این فرقی ندارد ممکن است در همه مرتبهای تر باشند

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

Exhibit 1(-)

PAPCO

$\sqrt{5}$

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 _____

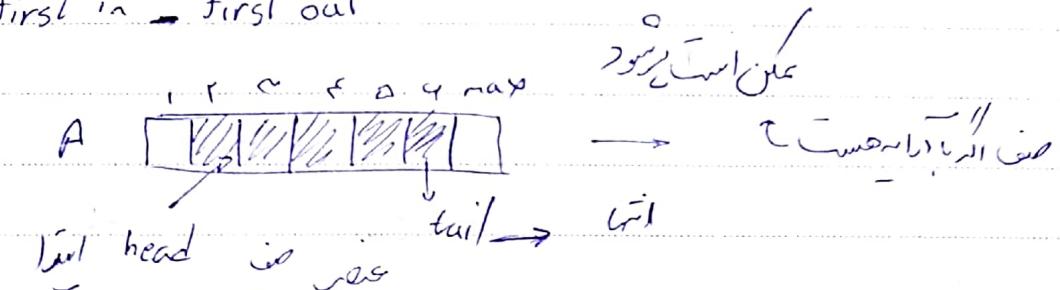
(V)

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
r if $a[i] == x$ then
c return true
e return false

is Empty(θ)

$O(1)$

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

is Full

$O(1)$

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

c else return false

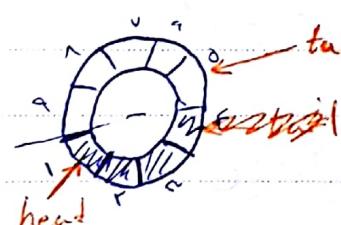
1. $\text{tail} = \text{head} = 1$
2. $\text{tail} + 1 = \text{head}$

Head:

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

: $\text{tail} < \text{head}$

tail = $\text{tail} + 1$: $\text{tail} + 1 = \text{head}$



tail: $\text{tail} = \text{head}$

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

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

is Empty(Q)

$O(1)$

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

else return false

is full(Q)

$O(1)$

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

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

c $t = 1$

r; if $t == \text{head}$ then

return true

else return false

BAPCO

④

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 $y \in S$
inflow

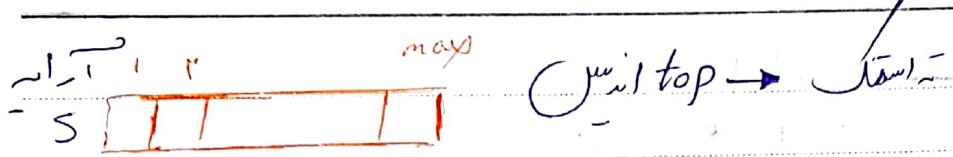
Get off in (out)

pop → stack $y \in S$
outflow

Get off stack $y \in S$

Subject

Date



$\text{push: } \text{top}++ \quad s[\text{top}] = x$

~~pop: $x = s[\text{top}]$~~ , $\text{top}--$

$\text{top} = \text{max}$

$\text{isEmpty}(s) \quad O(1)$

1. if $\text{top} == 0$

r. return true

c. else return false

q.

$\text{isFull}(s) \quad O(1)$

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

r. return true

c. else return false

q.

$\text{Find}(s, x) \quad O(n)$

1. for $i = 1$ to top do

r. if $s[i] == x$ then

c. return true

q. return false

$\text{Push}(s, x) \quad O(1)$

1. if $\text{isFull}(s)$ then

r. error

c. return

q. $\text{top} = \text{top} + 1$

2. $s[\text{top}] = x$

$\text{Pop}(s) \quad O(1)$

1. if $\text{isEmpty}(s)$ then

r. error

c. return

q. $x = s[\text{top}]$

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

3. 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 \rightarrow Prefix \leftarrow infix

taken

Postfix \rightarrow infix

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

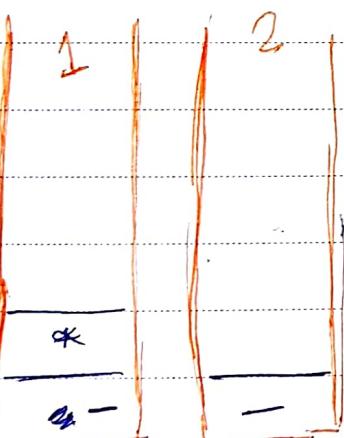
infix

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

infix



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



خواندن سه عبارت در خروجی نسبتی سود

خواندن سه عبارت بازدید کوتاه بازی از token

سینه در آن push می کند احری عبارت دری

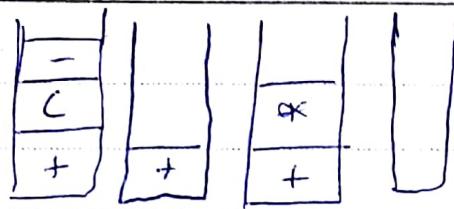


از سینه ادغام شرک نماید آن باید عبارتی شود

. pop از سینه شود

Subject _____

Date _____

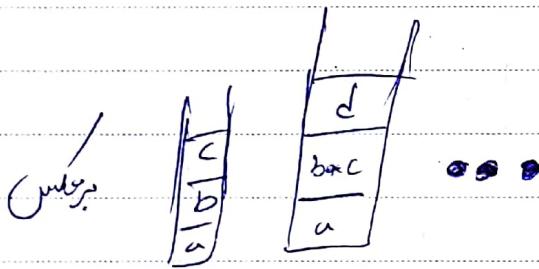


برانتر باز همراه push خواهد بود. برانتر سه قریب
است آنکه برانتر باز همراه pop شود.

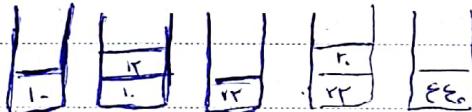
و در قریب اینجا فیل توکن خواهد نوشته شد.

بلی لرزیای خود را stack باز کن (ساده) خواهد داشت.

بلی علوفه خواهد داشت.

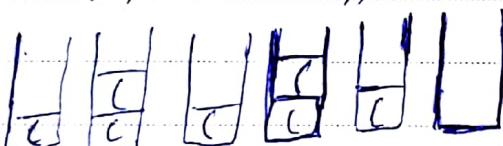


(a) $1 \cdot 11 + 1 \cdot *$

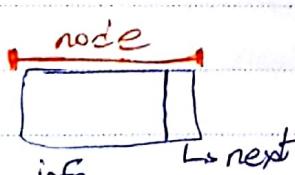
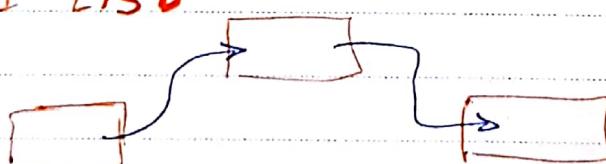


$((\alpha+\beta) - (\gamma * \delta))$

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

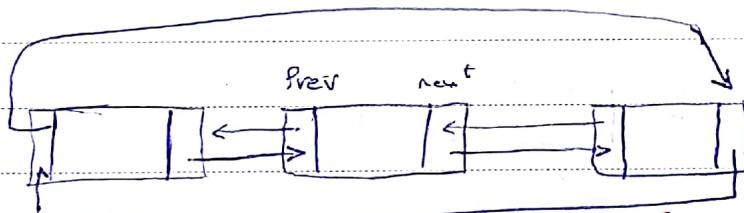


Prev

next

next

آخر عقدة في القائمة



Prev

next

آخر

Node * getNode() → إرجاع عقدة من القائمة (I)

```
{ Node *p;
```

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

```
return p;
```

```
}
```

```
void main()
```

(A) { Node * Ptr;

```
    (*Ptr).info = Ptr → info = 2;
```

```
    ptr → next = null;
```

P4PCO

Subject _____
Date _____

Node * p = first;

او میکنیم

Find & O

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

first



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

اگر info == 11 اینجا node میخواهیم

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;

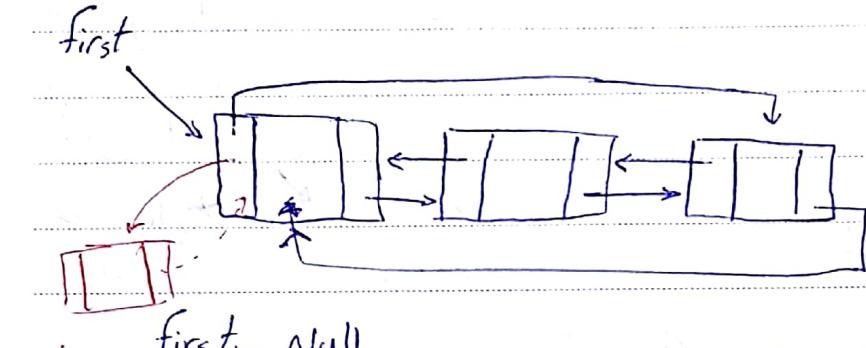
free (q);

}

P = P->next

}

لیست یوونگی در طرفه علوفی :



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 = a

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)

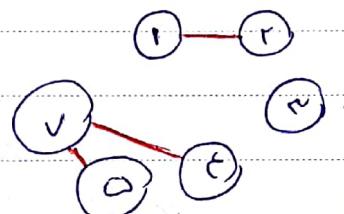
ساختار داده درخت

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

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

$$E \subseteq V \times V$$

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



گراف چند گزینه از هر رأس بهتر اس دیگر یک سری توان نیست / خوب نیست

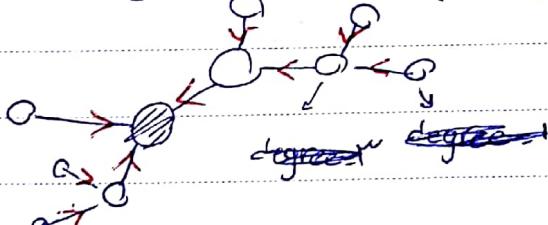
گراف دار گراف بیرون گشت



درخت یا گراف چنین است که قادر در باشد

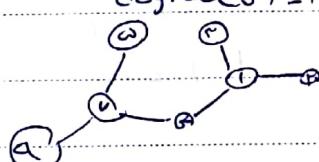
درخت دیگر: درخت دیگر را متفق گرام داشت درخت را رس (از خود یا دیگر را رس

نهایی یا root درخت را رس آن را به سمت



درجی راس: Node degree

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



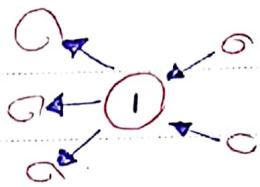
و ۱ حساب نماید

Subject

Date

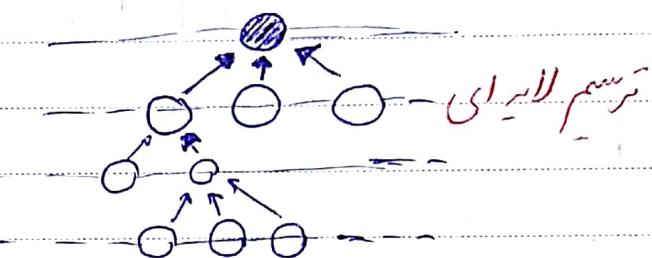
indegree = دخلی درجه (سویه)

outdegree = خارجی درجه (سویه)



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

بعد ادالہ کی تعریف

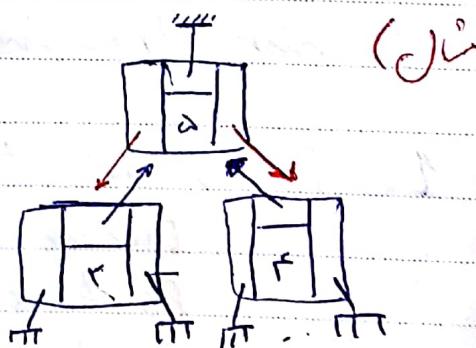
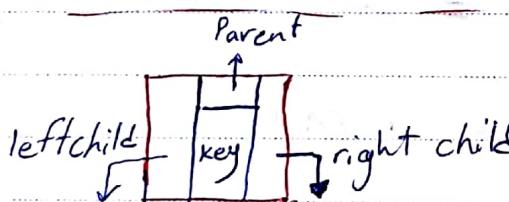


$h(v)$ ماکسیمیم (تاریخ) (height) ارتفاع

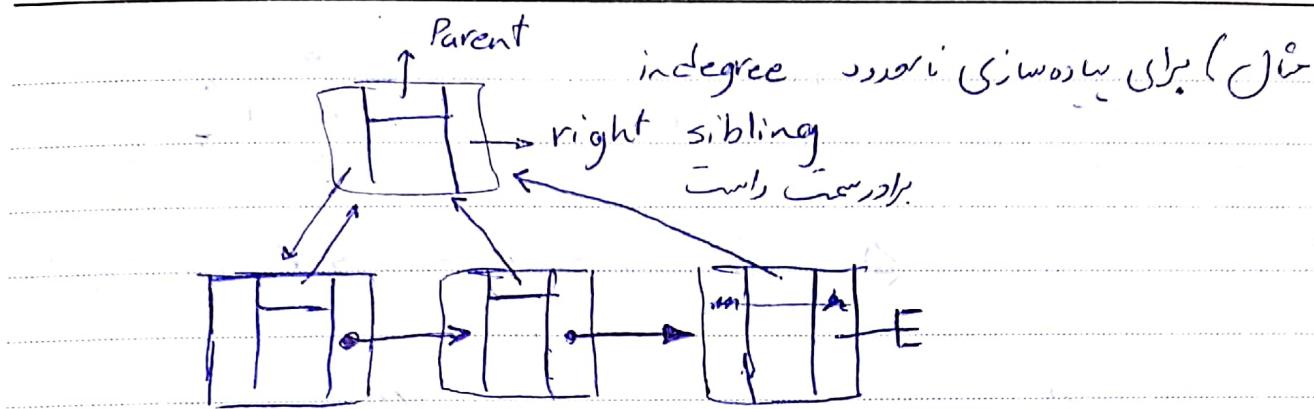
عوچ (depth) ناصلہ

ارتفاع درخت

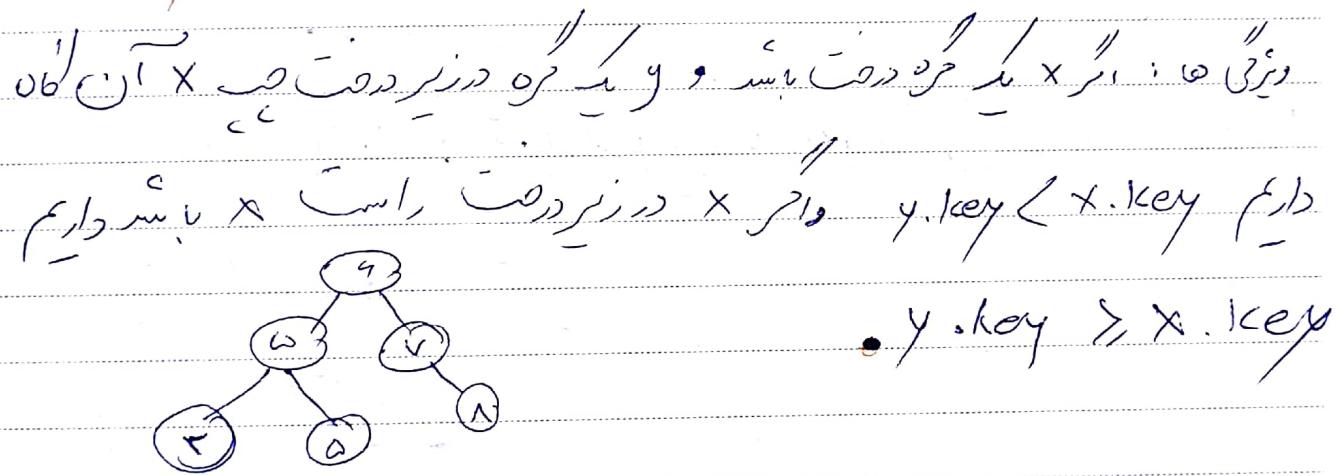
ایک جزوی درخت کو جو اپنے پڑیں



Subject _____
Date _____



Binary search tree (BST):



لورڈ جوں جیسے ترتیب = ترتیب in-order (جیسے)

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

1. if $x \neq \text{Null}$ then
Inorder-Tree-walk($x.\text{left}$)
2. Print $x.\text{key}$
3. 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 _____

مسار ایتریوی 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

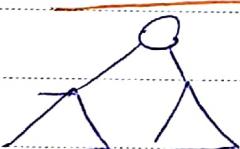
Tree-Min(x)

$\Theta(h)$

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

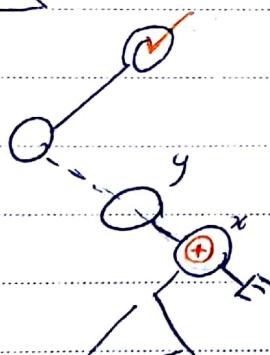
2. return x

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



SUCCESSOR

مره بالا فی صورتی



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

$x.\text{right} = \text{null}$: پس جی
 $y = x.P$

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

$x = y$

$y = y.P$

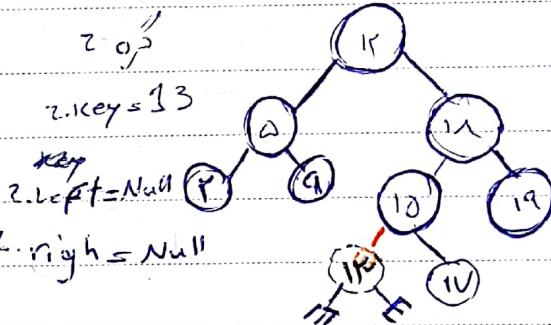
Subject _____
Date _____

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)

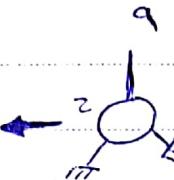
1. $y = \text{Null}$

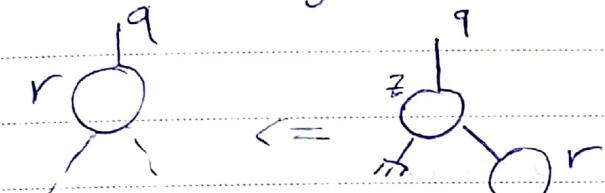
2. $x = T.\text{root}$

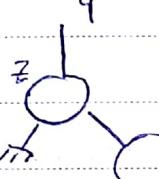
3. while ($x \neq \text{Null}$) do
 if ($z.\text{key} < x.\text{key}$) then $y = x$
 $x = x.\text{left}$
 else $x = x.\text{right}$
8. $z.P = y$
9. if ($z.\text{key} < y.\text{key}$)
10. $y.\text{left} = z$
11. $y.\text{right} = z$

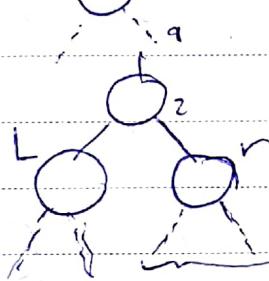
Subject _____
Date _____

Delete خفف

حالت ۱: $q.left = null$ و $q.right = null$ 
حذف فرزندی ندارد، کوچکترین فرزندی را بخواهد.

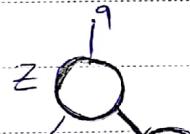


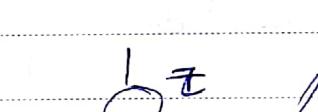
حالت ۲: $q.left \neq null$ و $q.right = null$ 
فرزند بزرگتر است: کوچکترین فرزندی را بخواهد.



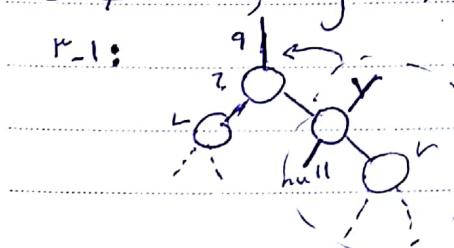
حالت ۳: $q.left = null$ و $q.right \neq null$: 
بزرگترین فرزند را بخواهد.

عن دردی min
نهایی خواهی

حالت ۱: $z.left = null$ 
بزرگترین فرزند را بخواهد.

حالت ۲: $z.left \neq null$, $z.right = null$ 
بزرگترین فرزند را بخواهد.

حالت ۳: $left \neq null$, $right \neq null$ 
نیزدست و جای نیزدست را قرار دهید و



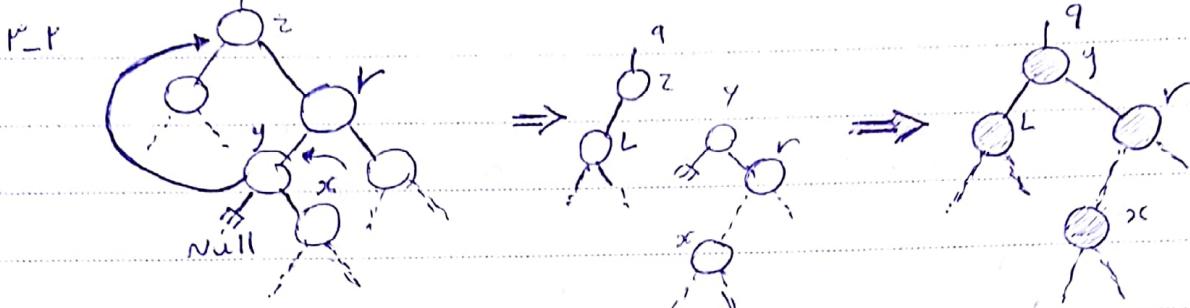
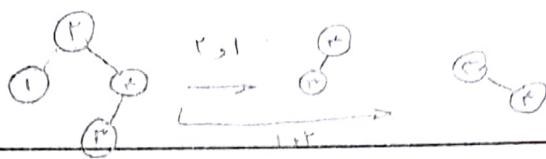
نیزدست و جای نیزدست را قرار دهید و

$y.left = L$

(نیزدست سمت چپ خیزندراست چیزی)

Subject

Date



Transplant(T, u, v)

$z \rightarrow u$

Delete v

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

2. $T.root = u$

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(~~Transplant~~ $T, z, z.right$)

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)

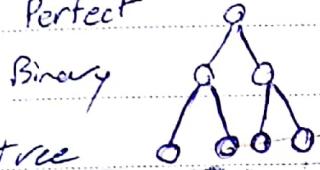
11. $y.left = z.left$

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

Heap (جعبه)

Complete (Binary) Tree جعبه (باينري)

Perfect



Complete

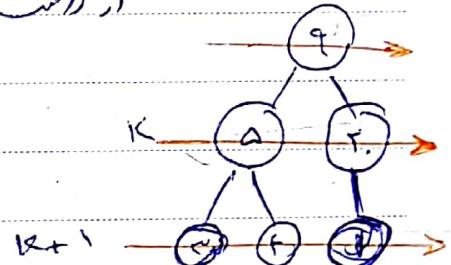
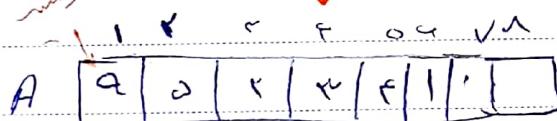


Perfect = درخت باينري
معنی هر چهار فرزند (فرزندان) یکی است

فرزندان درخت باينري
از راست به سمت ا 左 از ا 左 را حذف کرد

Heap

max heap = هر کدامیک از فرزندان بزرگتر از خود باشد



بزرگی بزرگ

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

Heap داشتاری Heap_size[A]

این جعبه: length[A]

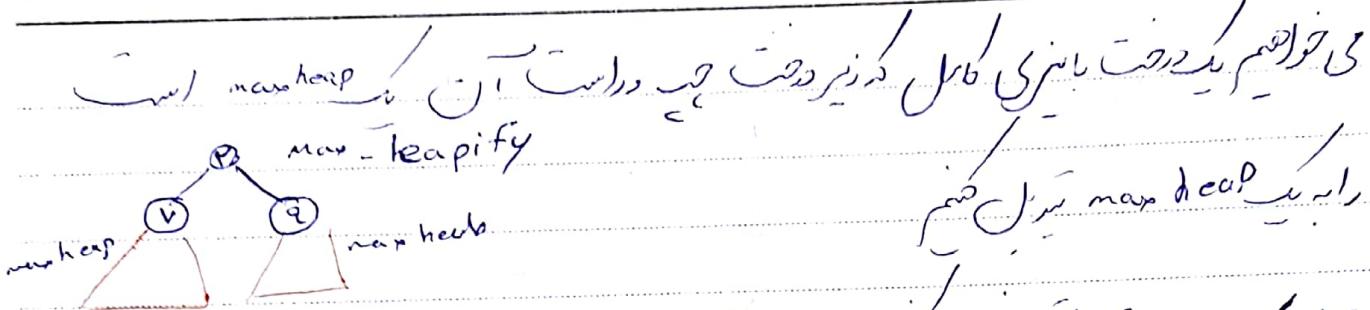
i > heap_size[A] $i > \text{heap_size}[A]$ برای هر چهار گره

این را درست برگزینید

عمر بیش از $\lceil \frac{n}{2} \rceil$ نیست

Subject

Date



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

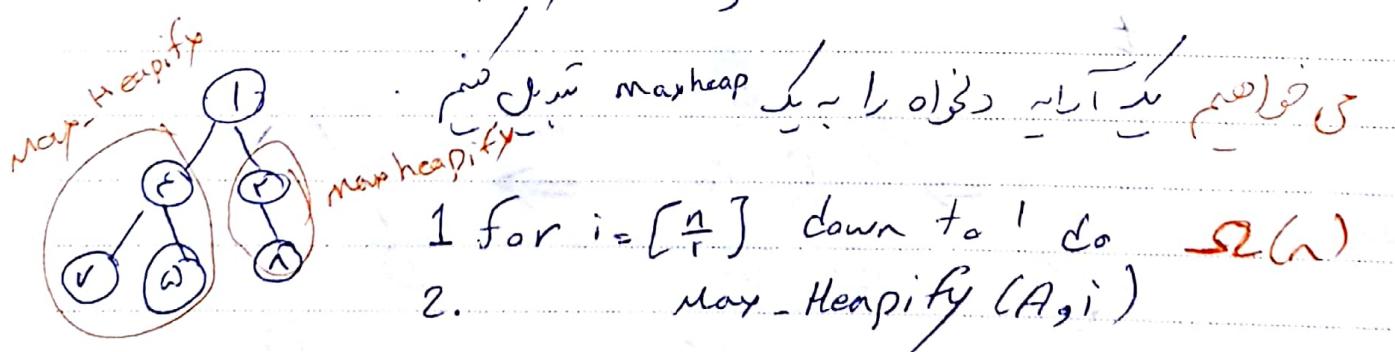


Max_Heapify(A, i)

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

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

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



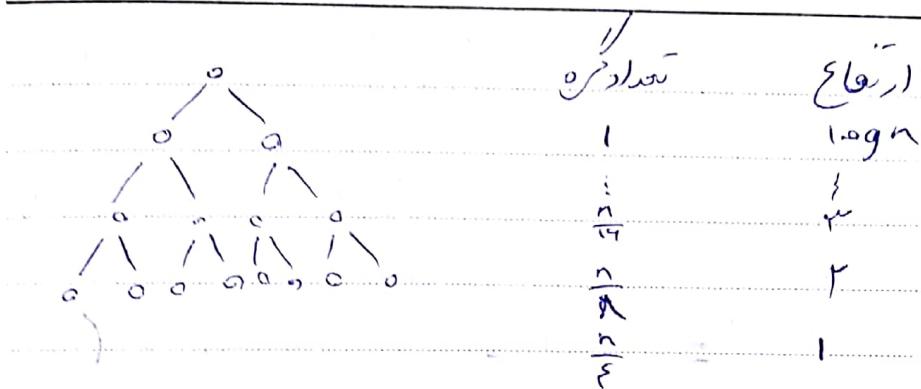
$O(n \cdot \log n)$ tight

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



Subject

Date

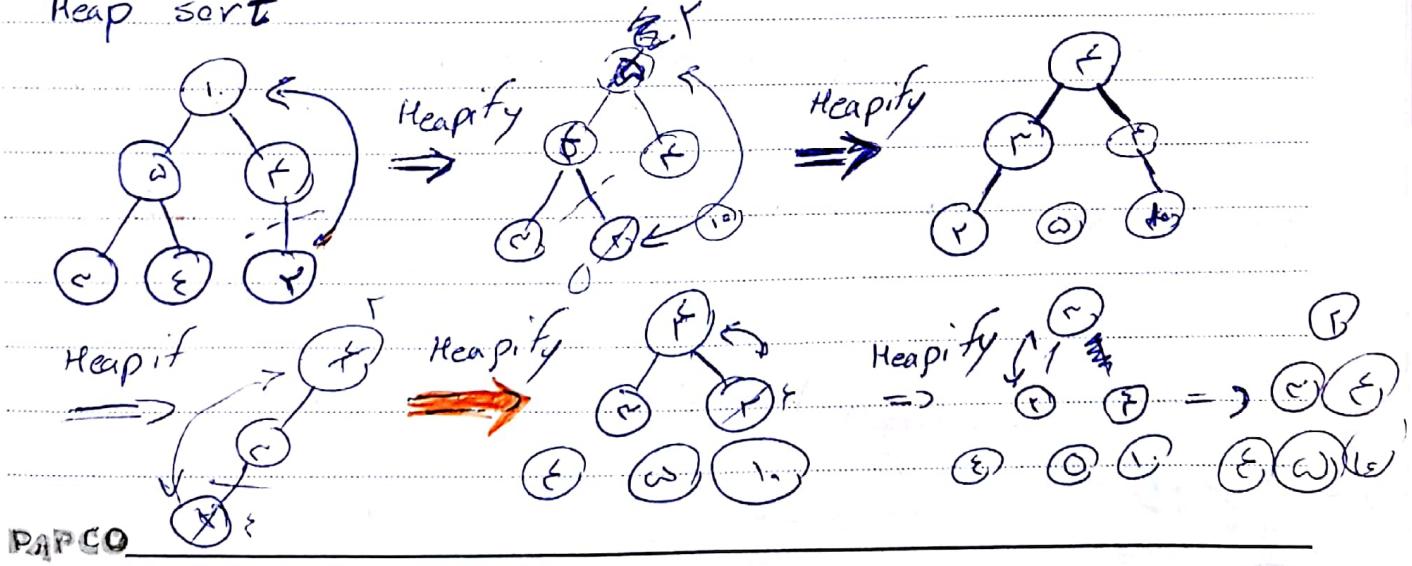


$\text{Pr}_{\text{avg}}: \mathcal{O}(1 + \frac{C}{n} + \log n)$

$$= C \sum_{i=1}^{\log r} \frac{n}{r^{i+1}} \cdot i^i = \frac{Cn}{r} \sum_{i=1}^{\log r} \left(\frac{1}{r}\right)^i \cdot i^i = O(n)$$

$$\sum x_i = \frac{x}{(1-x)^r} = Q_{(n)}$$

Heap sort



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 سندھت سادھے است

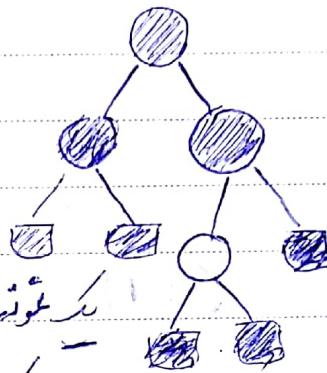
③ اگر گرہی دویتی دارد تو هر گرہی دویتی دارد

④ برای هر گرہی دویتی سرھی دویتی داری تعداد مساوی گرہی دویتی دارد

Subject _____

Date _____

Null = بیک
Null جو = بیک جو



جواب = خاسو

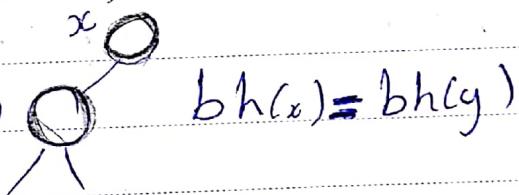
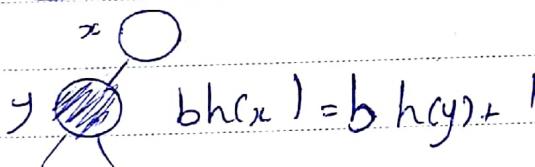
Red-Black ایجاد کردن

کل سریع ترین برای طول بسته دستگاه

black height = bh(x)

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

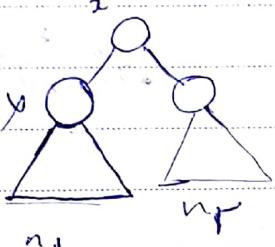
(جواب عرضی)



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

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

Subject _____
Date _____



الثوابت المطلوب إثباتها هي $n \geq r^{bh(x)} - 1$ و $n_l \geq r^{bh(y)} - 1$ و $n_r \geq r^{bh(z)} - 1$.

أولاً: $n \geq r^{bh(x)} - 1$

لأجل ذلك $n \geq r^{bh(x)} - 1 \geq r^{bh(x)-1} + 1$

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

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

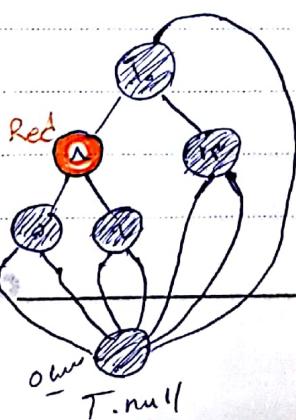
~~$n \geq r^{bh(x)}$~~ $\geq r^{bh(x)} - 1$

$h=0 \rightarrow n=1$ لآن $r^0 = 1$

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

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

• AVL tree - search \leftarrow Red-black Tree



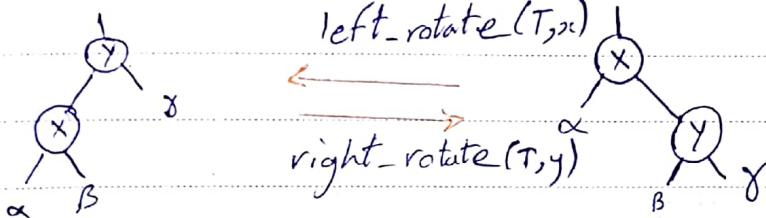
١٤

٣٦

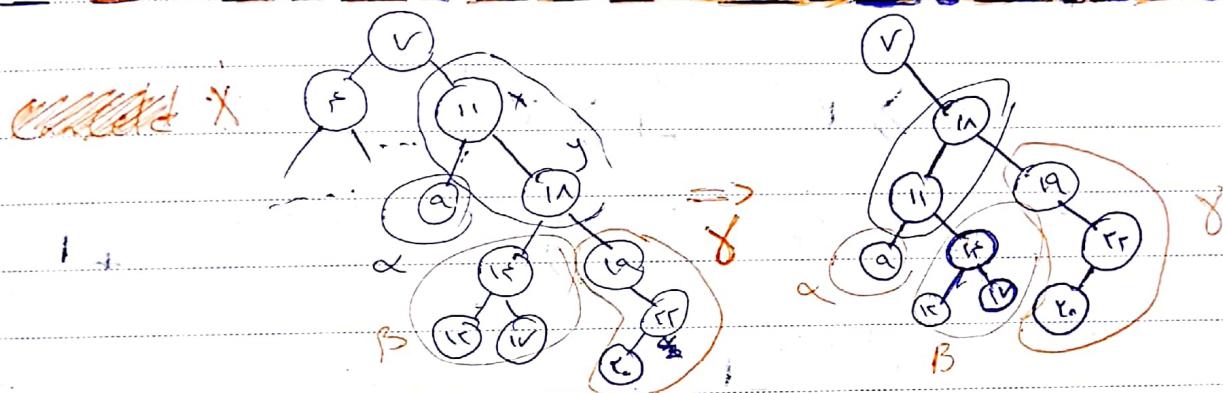
الرجوع إلى
الخلف

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

Rotation



• تغيير عقدة



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)

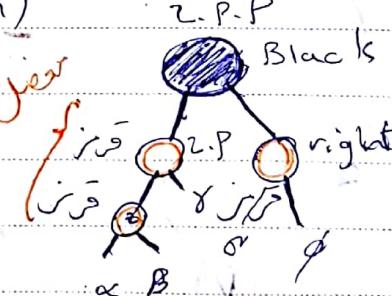
initial red-black b. node by

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)

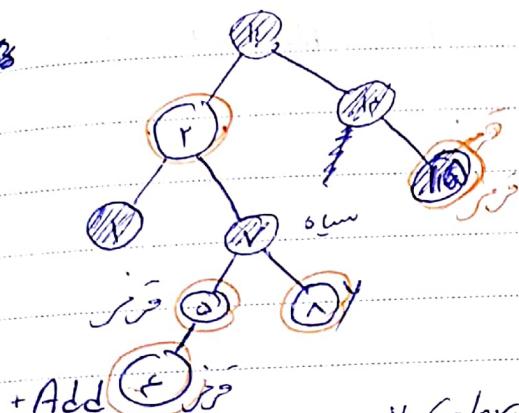
→ RB-Insert-Fixup(T, z) $O(\lg n)$

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 دینف نظر سلسلت.

RB



: Case 1

حیزاست y , $y = y.p.p.right$

برای حل مشکل بزرگ شدن درخت باید

+ Add

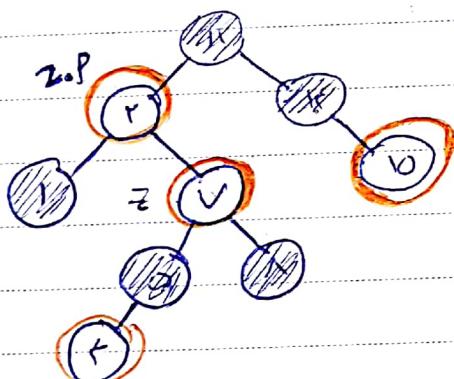
$y.color = \text{Black}$

$= z.p, z$

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

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

$\underline{z} \quad z = z.p.p$



: Case 2

حیز است y , $y = z.p.p.right$

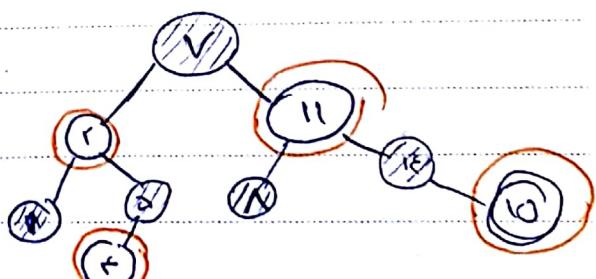
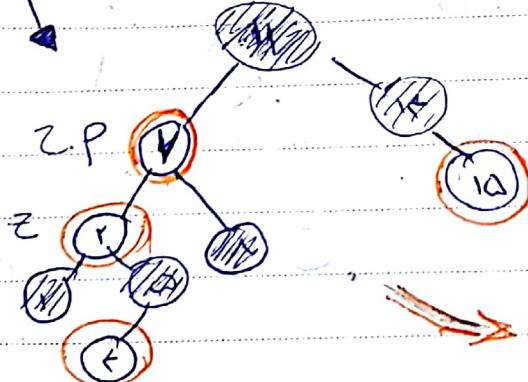
برای حل مشکل بزرگ شدن درخت باید $z.p$ را



: 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

loop invariant:

یک گزاره که در طول چنین چنین دستورات است است.

بررسی محتوی علی چنین Red-Black

loop invariant

ا) حروف زیر قرمز است

ب) اخر دو حرف Z.P زیر است آنها Z.P زیر است

c) اخر دو حرف Z.P زیر است آنها Red-black (تفصیل نمایند) و دویتی را تفهی فرموده باشند

امروزی در (تفصیل نمایند) این است در جریان است و قرمز است. اخر دو حرف Z.P را تفهی فرموده باشند

د) دلیل این است در جریان از زیر قرمز است

حالت اولیه (اگر اول)

بینهایت کمتر و بینهایت بزرگ داشته باشی آدرسی که از خارجی while دستور است

عمل از فرایانی P ، Fixup ، Color-Red

a) بینهایت درست است b) اخر P حرف را رسیده باشد چون Z.P زیر داشته باشد زیرا دو حرف Z.P رسیده بوده رسیده است اخر P حرف رسیده باشد زیرا دو حرف Z.P رسیده باشد

با انتها همچشم نیست 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 درجه سیرها بعد از خواهی ساده می باشد ساده است (دریجی دفعه

می شود). دریجی دفعه عربی است دفعه نی شود و در عربی دفعه فرز است

شما خواهی دفعه است دریجی دفعه شد این است در رسمیت

شما خواهی دفعه شده است دا در عبارت داشت دفعه شد عین در P. 2 به این

(فرز زدن T null است دفعه است) در صورت دفعه فرز یا سی فرز می شود

و از دفعه شد بـ P فرز شد این دریجی دفعه می شود و بالعكس

Termination کهی که () است

دریجی ①، ②، ③، ④ فرز نی شود اگر دریجی ④ فرز شد دلیل این است در ترس

است فرز است در این صورت دستور T بـ از طبق while

عمل را حل دعاء دریجی های برقرار است.

اگر دریجی ④ فرز شد دلیل این است در وقف فرز است. دفعه حلقه ای while

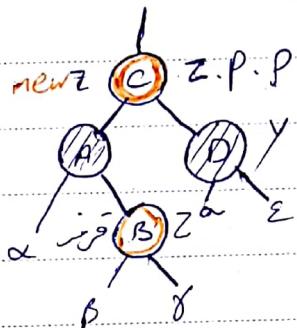
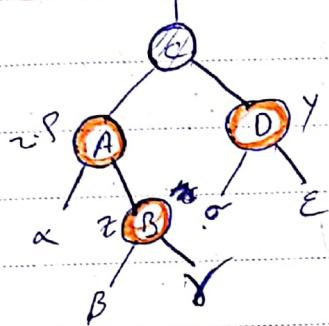
می شود و P سی و است سی خواهد دفعه می شود.

maintenance

Split Case

آخر ای داده ای بزرگتر while اینست ای داده ای بزرگتر

Case 1:



آخر ای داده ای است ایست کریز Z.P.P و newz داده ای است ایست کریز Z.P.P

دایرکت دایم (Z.P.P و دردار)

آخر ای داده ای است ایست کریز Z.P.P

آخر ای داده ای است ایست کریز Z.P.P و دیگر داده ای است ایست کریز Z.P.P و دیگر داده ای است ایست کریز Z.P.P

آخر ای داده ای است ایست کریز Z.P.P و دیگر داده ای است ایست کریز Z.P.P

آخر ای داده ای است ایست کریز Z.P.P و دیگر داده ای است ایست کریز Z.P.P

آخر ای داده ای است ایست کریز Z.P.P و دیگر داده ای است ایست کریز Z.P.P

آخر ای داده ای است ایست کریز Z.P.P و دیگر داده ای است ایست کریز Z.P.P

اگر ویرجی (۱) برقرار نہیں ہے:

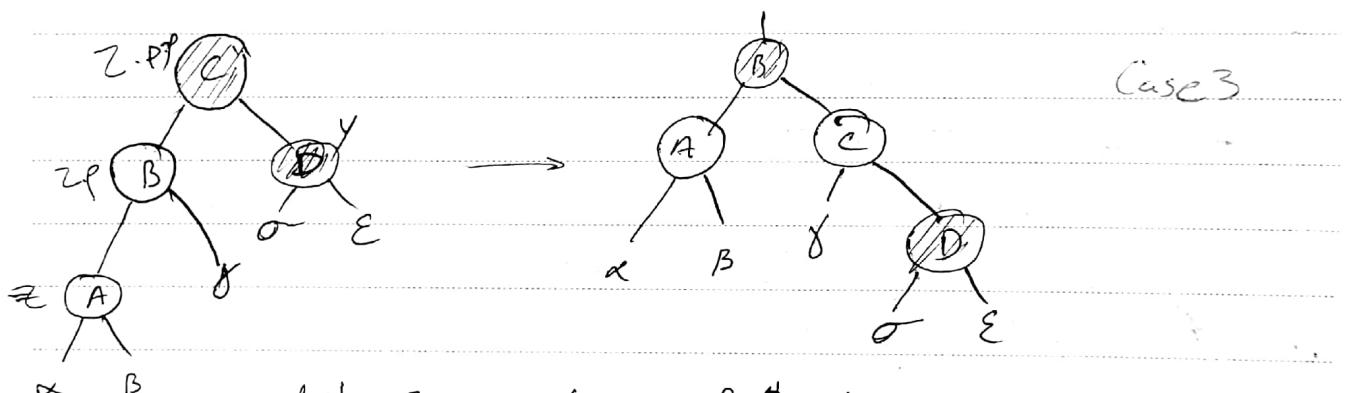
بچھے بھرائی حرد Case 1 دوستی کی صورت میں داد دینے والے افراد

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 اگر

کوئی ویرجی (۴) میں انت تھن سر (اصحی کر پا جائیں گے)



while Z.p.color == Red do

{

if قرمز

Case 1

else if فربنا

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~~ $v.p = u.p$

RB-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. RB-Transplant($T, z, z.\text{right}$)
6. elseif $z.\text{right} == T.\text{null}$ then
7. $x = z.\text{left}$
8. RB-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 // RB-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.

$x = T.root$

X-

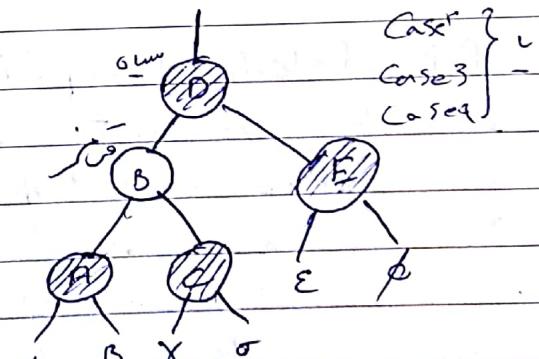
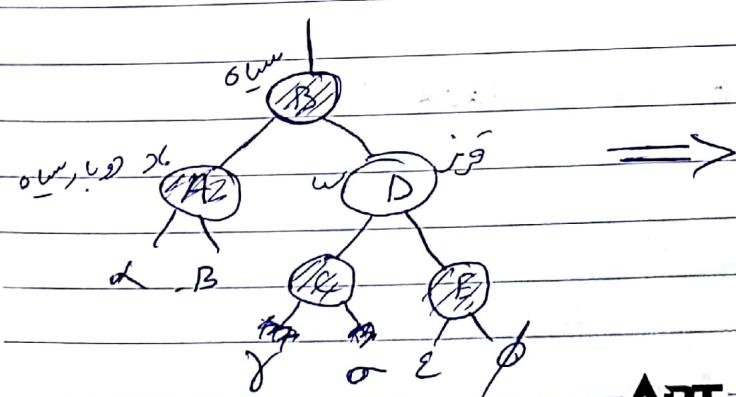
Y-

Z-

else (same as then -- left → right)

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

Case 1

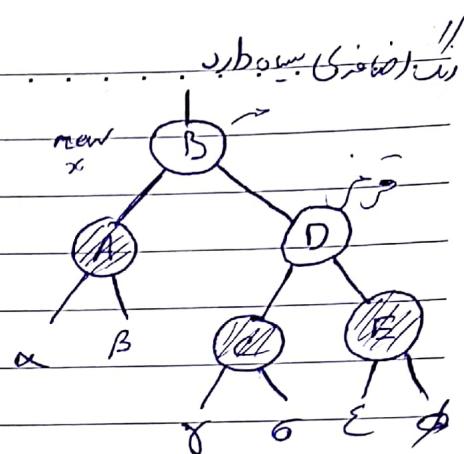
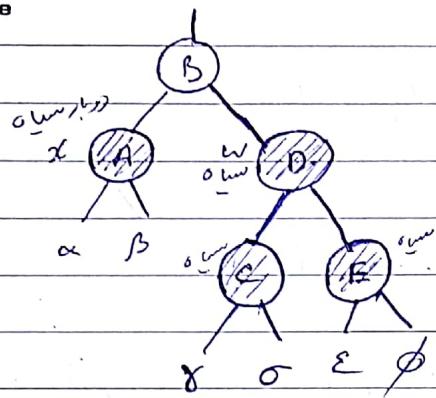


ART

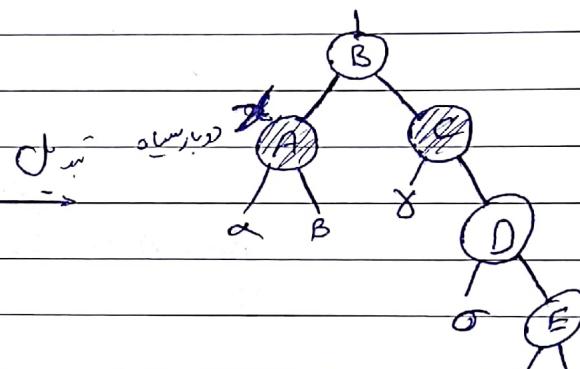
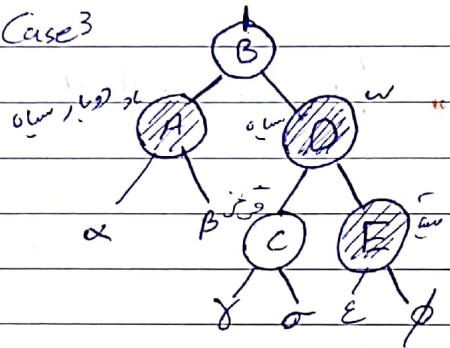
Subject

Case 2

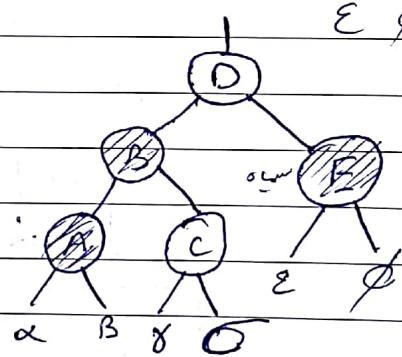
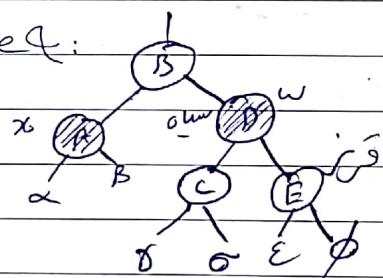
Date



Case 3



Case 4:



Augmenting Data structures

اده های سریع برای این داده های افزودن و حذف امکان دارند

Red Black Tree

جیلی

ART

۴۲

Subject

Date

then (lower) \rightarrow (higher) \rightarrow selection

ex. (just x) rank in order: ranking.

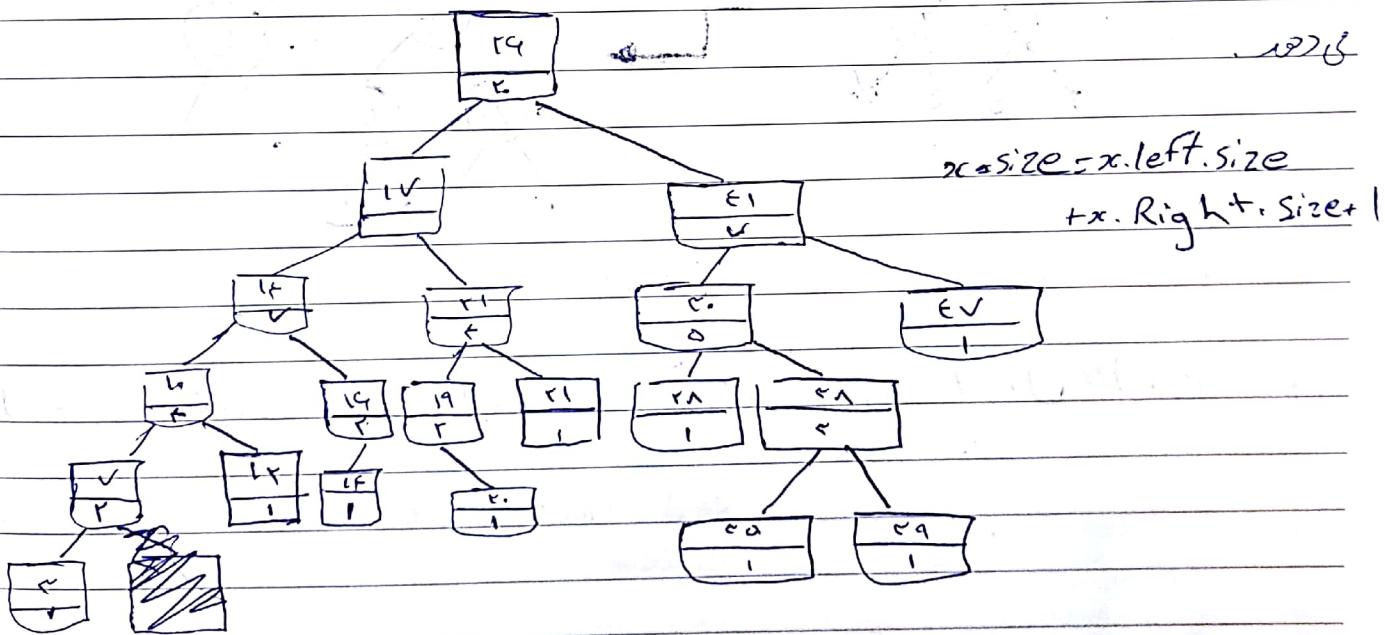
بامداده (نیز) سری دستان (O(nlogⁿ)) حل یافته

مودرلیو^ن (O_n) حمل جردن

Red-Black n- selection using J

$\omega \omega$, $\mu^+ \nu$ $k=10$

جزئیاتی مورخی جزئیاتی size (اضافی) نمایند از جزو های زیر دست برسید و در اینجا



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

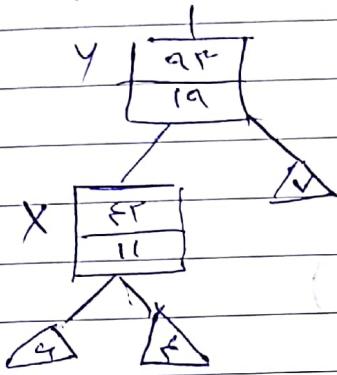
(e)

Subject

Date

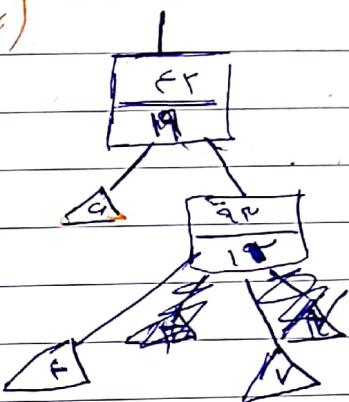
١) ایجاد: از جمله حجت های جبرین است - size = $O(n^2)$ (lagr).

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 $\{ y.size = x.size \}$

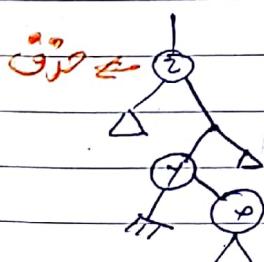
$$\text{left-rotate } x \quad | \quad x.\text{size} = x.\text{left.size} + x.\text{right.size} + 1$$

مکاری عن جنف

۱) ای حصہ: زحل اور سیاروں کا حصہ (ترکیبی حصہ سہ) ترکیب از

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

O₂, Ozone; Fixup



Subject:
Year. Month. Date. ()

Interval Tree

ein Intervall $[t_1, t_r]$

$$\{x \in R : t_1 \leq x \leq t_r\}$$

obj i.low = t, i.high = tr

initial seeds (جذور) overlap in some

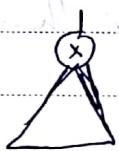
حالت

لارڈ ایڈوارڈز

(i. high / low) تُعْتَدِي (r) تُعْتَدِي (s)

(i.low > i.high) تاپی تاپی (۳)

x opengl x.int . (v) Red-Black Tree (Implementation)



میں ترکیب سے حاصل ہوتے رہتے ہیں

بازه های آدرزیر دست برس اول اول

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

x.key = x.int.low

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

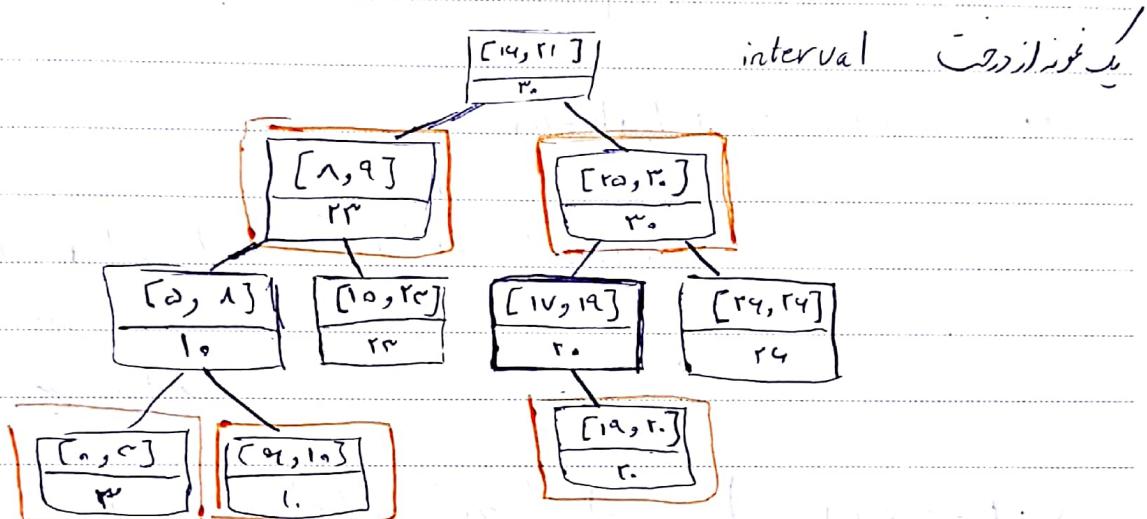
Interval-Insert(T, x)

: 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 نهاده جشن باشد
while دستور طرف دار
که در زیر دست باری خود را نزدیک دارد.

عمل (از اجرای حلقاتی) Initialization while (حالت پایه)

بردست باری درست است که نیز مثل درست تا است

حالت تغذیه maintenance :

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

نایت حلقات درست است.

حالت کامیابی (1) اگر صحت درست نباشد else درست نباشد (5 بند)

در درست صورت باندهای طلب در زیر دست left میباشد دوامز max

چنانچه و جو در دست باری در زیر دست راست است (س) نایت حلقات را فقط

بازهای در زیر دست حبیب باشند high

i. low < x.left.max = i' high if برقرار باشد: (2) در حالی

Subject

Date

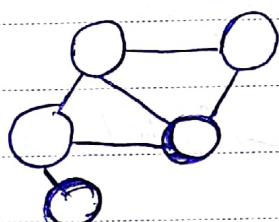
اف) $i \leftarrow i \cdot low < i \cdot high$ باره بازدید و تحسیل طرد.

ب) $i \leftarrow i \cdot high < i \cdot low$ بازدید ناصیح بازدید از زیردرخت راست اسراک ندارد.
لئے هر زده از زیردرخت راست

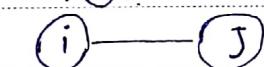
$G = (V, E)$
مجموعه ماله ها
مجموعه رؤس

الترمیم جای حرف

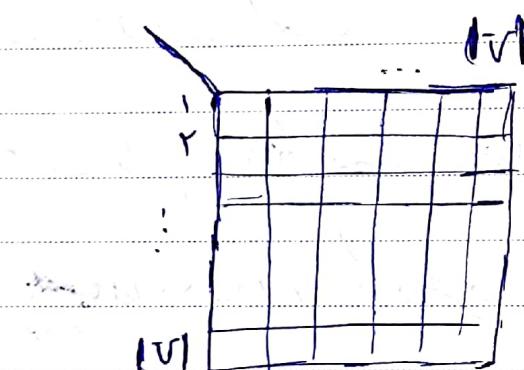
$$E \subseteq V \times V$$



دور بست



کوت در



$$\text{ماتریس مجاہلی} = \Theta(V^2)$$

ماتریس مجاہلی

$$A[i, j] = \begin{cases} 1 & \text{اگر زیرگیل ندارد} \\ 0 & \text{اگر زیرگیل دارد} \end{cases}$$

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

Subject

Date

~~100~~ 1U, 1U-1

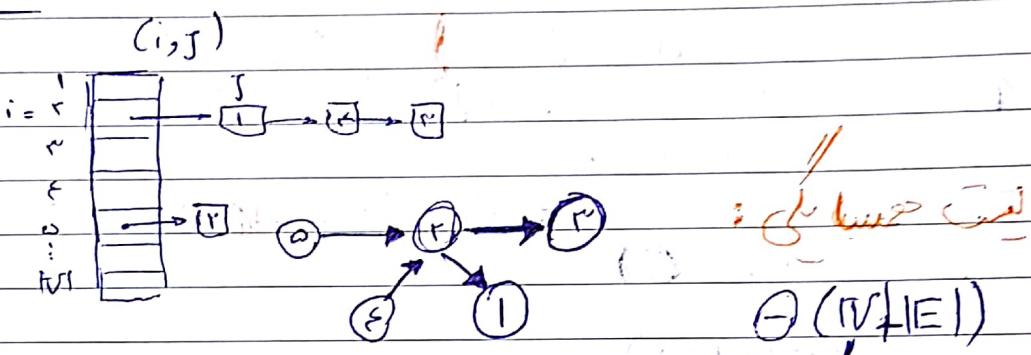
حمد لله رب العالمين (عجايب العجب) دار

جبل

$$\sum_{j=0}^{O(i)} A[j, j] = A + (i-1) \cdot |V| + i - 1$$

Fiji

Förber



بفرض $\sum b_n$ لون

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

\downarrow

مقدار

جواب: sparse

سیاسی حکومت

جستجوی اولیه مردگان (Death First search) DFS

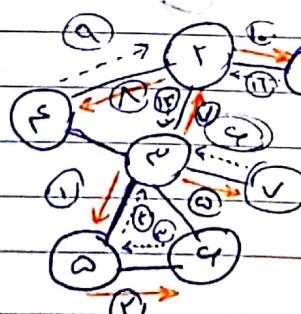
لیگان روی سنت (

) BFS

١١

تومر حرم ای / حمسی بر دیر چشم به ای نه

اکریسٹن دری بحر جسے میری visit



ART

Subject

Date

حروف // Σ (r_i)

$E \text{ degree} = \Theta(|E|)$

DFS (G, V)

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

1. visited [v] = true $\rightarrow \Theta(|V|)$

2. for each w that is neighbour of v do } $\Theta(|E|)$

3. if visited [w] == false then

4. $\text{DFS}(w) \rightarrow \Theta(|V|)$ (جزئی خرڅان)

Depth-First-search (G, V)

1. $\text{DFS}(G, V) \quad \Theta(|V| + |E|)$

2. for $i=1$ to $|V|$ do

3. if visited [i] == false then } $\Theta(|V|)$

4. $\text{DFS}(i)$

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

جمل برای حروف از حروف های

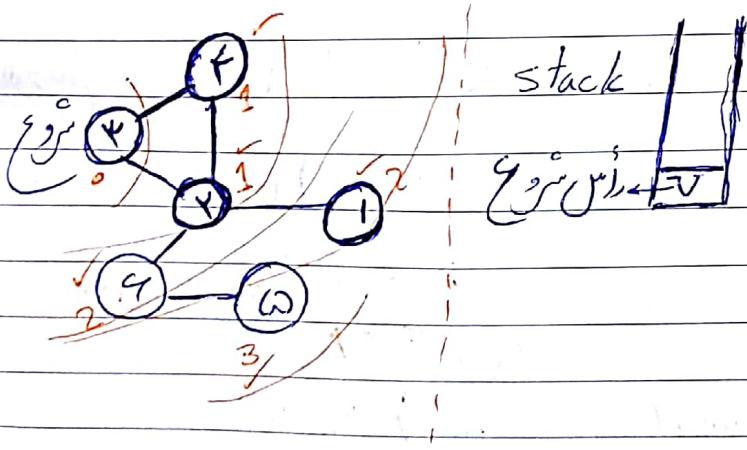
ساده لوری \uparrow $|V| + |E|$ order \rightarrow $|V| + |E|$ order

Amortized Analysis (لکل سرچ) (در دفعه تراست و جزوی نهایی)

BFS

: DFS

لوری غرباً نهایی



Subject

Date unRecursive - DFS (G, v)

۱۵

1. visited [v] = true

2. Create an empty stack S

3. s.push(v)

q. while s.empty() != true do

5. | ws: Find-next-unvisited-neighbour(S,top)

6.

7 if visited[w] = true

8. // s.push

$$\Theta(1 \vee |t| + |E|)$$

UnRecursive_DFS2(G,V)

1 Create an Empty stack S

2. S.push(v)

3. while ~~stask~~ s.empty() != true do

4. $w = s.pop()$

5. ~~visited~~ Ew7 = true K

Now for each i that is neighbour of w_0

~~8.~~ 8. push (n)

Subject

Date

56 6. if visited[w] ≠ true then → while job

67 7. visited[w] = true

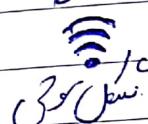
78 for u that is a neighbour of w do

8.

S.push

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

BFS



unRecursive_BFS(G, v)

1. Create an Empty Queue Q

2. Q.add(v)

3. while Q.empty() ≠ true do

4. w = Q.Del()

5. visited[w] = true

6. for each u that is a neighbour of w do

7. if visited[u] ≠ true then

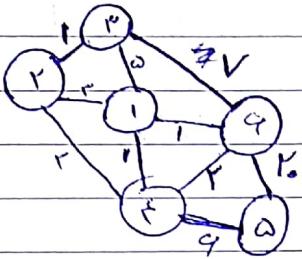
8. Q.add(u)

9. visited[u] = true $O(\text{size}^2) = O(|V| \cdot |E|)$

$$\text{Size}_{\text{new}} = C_1|V| + C_2|E| = \Theta(|V|+|E|)$$

Subject

Date



گرامر میر دریان

طبل سیر: جمع ذریں یا لہجے میں

الخوارزمي BFS

(بلک حراف های بعدی نزد)

Floyd - warshall

Dijkstra

all pairs

لِزَجْرَانِ

(دور بارز) حق خواه در اینسته نیست

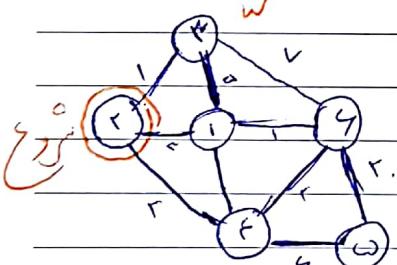
لـ جـ لـ لـ

وزیر نامہ متفق پائیں

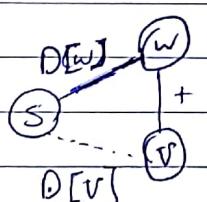
* اگر درستی دیگر دلست پایس دیر چوناں ترین مسیر دیگر نخواهد داشت.

Dijkstra الورقة

دنبالت طبل دنهان عجزه از این نوع دیگر \rightarrow خلاصه در



$$D[i] = \begin{cases} w[s, i] & \text{if } s \in \cup_{j=1}^k S_j \\ \infty & \text{otherwise} \end{cases}$$



$D \rightarrow \text{out}(\text{in}(D[w]))$

-ART

aa

طول کوتاه‌ترین مسیرهای w است و در ترتیبی ω اندیس

برای update $D[V]$ مسیرت را بروز w با v در قریب سازی کنیم.

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

$$\text{مقدارهای } w = \{1, 4\}$$

$$D[1] = \min \left\{ D[1], D[r] + w[c, r] \right\}$$

$$D[4] = \min \left\{ D[4], D[\infty] + w[\infty, 4] \right\} = \Delta$$

$$D[1] = \min_{r=1}^{\infty} \min_{c=r}^{\infty} D[c]$$

$$D[1] = \min \left\{ D[1], D[\infty] + w[\infty, 1] \right\} = \Delta$$

$$D[4] = \min \left\{ D[4], D[\infty] + w[\infty, 4] \right\} = \Delta$$

$$D[0] = \min \left\{ D[0], D[\infty] + w[\infty, 0] \right\} = \Delta$$

$$\text{جذر مجموع} = |V| - k$$

$$\text{جذر مجموع اندیس} = O(|V|)$$

۰۶۶

w مقدارهای c برای update

degree(w)

ART

$O(|V|)$

دسترسی سریع برای c

Subject

Date

١١
٦٦

$$\text{Time} = O(|V|) + \boxed{(|V|-r) \times O(|V|)} + \sum_w \text{degree}(w) = O(|V|^2 + |E|)$$

$$= O(|V|^2)$$

min heap جدول

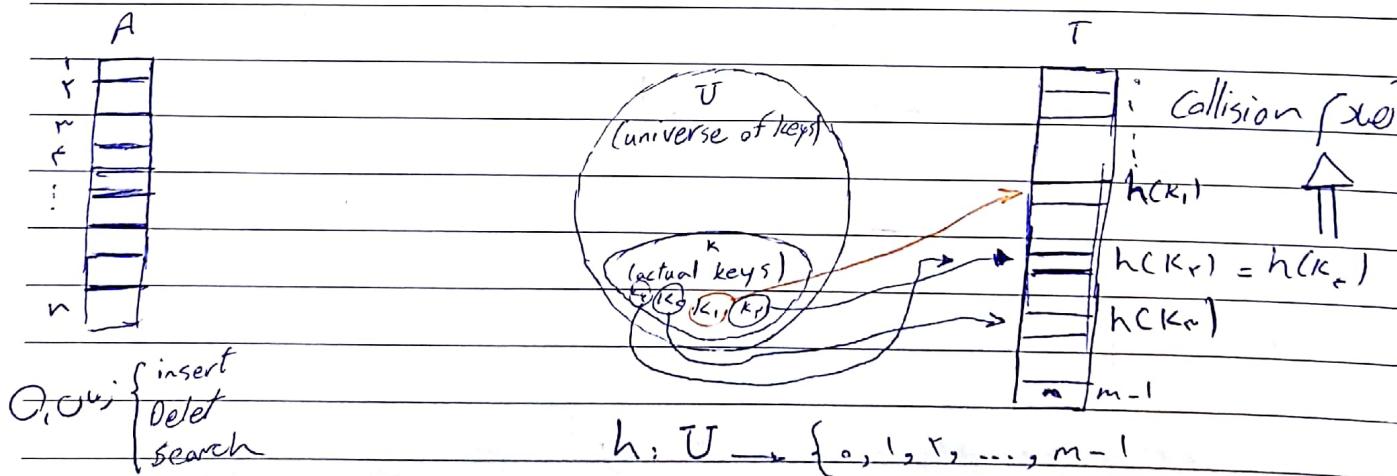
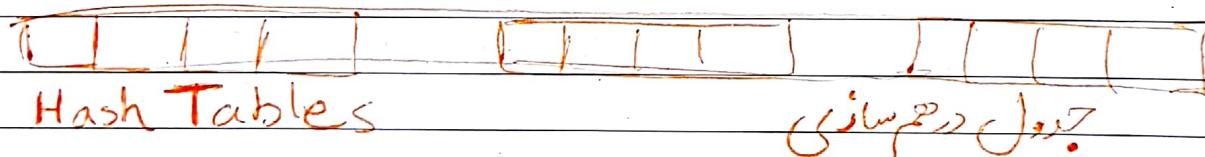
$O(|V| \cdot \text{min})$ حداقة $\in O(|V|) \leftarrow$ دفعه جدول

$O(\text{degree}(w) \cdot |V|)$ update
 $+ O(\text{degree}(w)^W)$ جدول
 $+ O(\text{degree}(w) \cdot \log |V|)$

$O(\log |V|)$ heapify, min خلق

$$\text{Time} = O(|V|) + O(10) + (|V|-r) \times O(1 + \log |V|) + \sum_w \text{degree}(w)$$

$$+ \log(|V|) \times \sum_w \text{degree}(v) = O(|E| \cdot \log |V|)$$



ART

AV

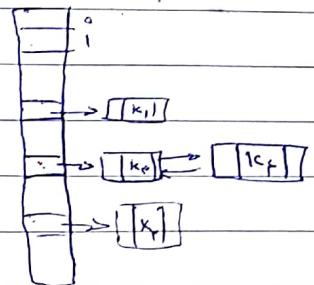
Subject

Date

in $O(n)$ $\Theta(n)$, $\Theta(1)$ $\Theta(k)$ $\Theta(1)$

linked list ~~linked~~ collision $\Theta(n)$

(\rightarrow slot) slot



Chained_Hash_Insert(T, x)

1. insert x at the head of list $T[h(x.\text{key})]$

Chained_Hash_Delete(T, x)

1. delete x from the list $T[h(x.\text{key})]$

Chained_Hash_Search(T, k)

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

جستجوی ایتم با کلید در لیست مخصوصی که با کلید k مرتبط است

نیز T یک لیست میباشد که از m لیست $T[1], T[2], \dots, T[m]$ تشکیل شده است

$$\text{Load factor} = \frac{n}{m}$$

$$n_j = |T[j]| \text{ عدد ایتم های لیست } T[j]$$

$$n = n_0 + n_1 + n_2 + \dots + n_{m-1} \quad E[n_j] = \alpha = \frac{n}{m} \quad (\text{یعنی})$$

Subject

Date

دسمبر چینی جدول دفعه ای سیمین و پنجمین باره (۱۱-۱)

سازی میزبانی ساده "نطایج جدول دفعه ای سیمین و پنجمین باره" (۱۱-۲)

$$\text{سازی میزبانی ساده} = \sum_{i=0}^{m-1} \frac{1}{m} \times (\alpha + 1) = 1 + \alpha$$

دسمبر چینی جدول دفعه ای سیمین و پنجمین باره (۱۱-۲)

سازی میزبانی ساده "نطایج جدول دفعه ای سیمین و پنجمین باره" (۱۱-۲)

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

$$x_{ij} = I\{h(k_i) = h(k_j)\} = \begin{cases} 1 & h(k_i) = h(k_j) \\ 0 & \text{otherwise} \end{cases}$$

• سازی میزبانی ساده سیمین و پنجمین باره

$$k_i = x_i \cdot \text{key}$$

$$n_{h(k_i)} = \sum_{j=1}^n x_{ij}$$

$$h(k_i) = h(k_j)$$

$$E[x_{ij}] = 1 \times \Pr\{h(k_i) = h(k_j)\}$$

$$+ 0 \times \Pr\{h(k_i) \neq h(k_j)\}$$

$$= \Pr\{h(k_i) = h(k_j)\} = \frac{1}{m}$$

$$\Rightarrow = 1 + \frac{1}{n} \sum_{i=1}^n \frac{n-i}{m} = 1 + \frac{1}{nm} \times \frac{\alpha(n-1)}{r} = 1 + \frac{n-1}{nm} \times \frac{\alpha+1}{r}$$

Subject

Date Hash functions

$$h: U \rightarrow \{0, 1, \dots, m-1\} \quad h(k)$$

سیگنال توسع (حکم طبقاً سیگنال) بست اما حکم این توابع متغیر باشد. مثلاً باید

باید هم باید هم در طبقه ای را از اعداد حقیقی حاوی در بازه $[K]$ به صورت متوالی

فرض داشت $h(k) = \lfloor k \cdot m \rfloor \text{ mod } n$

و بنابراین سیگنال طبقه ای داشتم که سیگنال توتابع در حوزه ای انتگرالی تفرقی نمایم

در حوزه ای پذراحت نباشد و در حوزه ای نباشد.

رشی دفعه سازی

$$h(k) = k \text{ mod } m$$

multiplication

رشی دفعه سازی

$$\langle A \rangle_1 = A \text{ mod } m$$

$$h(k) = \lfloor m \cdot (k \cdot A \text{ mod } 1) \rfloor$$

$$A \sim (\sqrt{m} - 1)/r = 1418.3398 \dots$$

$k \cdot A$ دفعه سازی

universal Hashing:

Subject

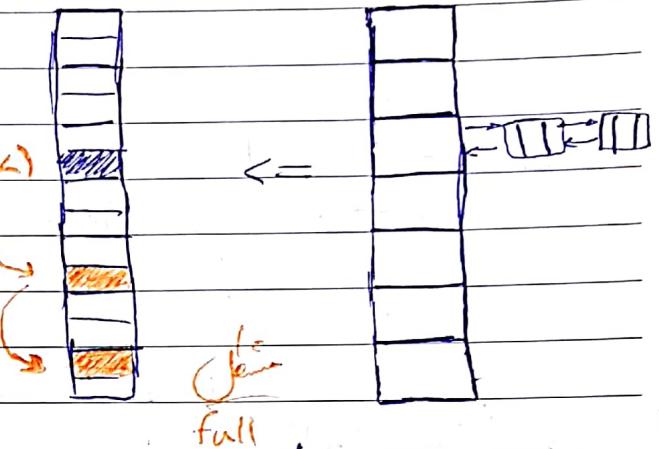
Date

open addressing collision

جزءی از داده را در خود نداشت

جایگزینی Add

جایگزینی جایگزینی بعدی با هم offset



Probe sequence

Probe

$\langle v, i, r, \dots \rangle$

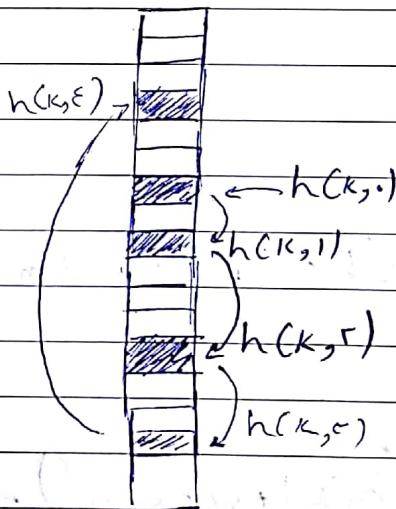
$$h: T \rightarrow \{0, 1, r, \dots, m-1\} \rightarrow \{0, 1, r, \dots, m-1\}$$

(Sequence)

$\langle h(k,0), h(k,1), h(k,r), \dots, h(k,m-1) \rangle$

Hash-insert(T, k)

1. $i = \phi$
2. repeat
3. $j = h(k, i)$
4. if $T[j] = \text{Null}$ then
5. $T[j] = k$
6. return j
7. else $i = i + 1$
8. until $i = m$
9. error "Hash Full"



ART

Subject

Date

Hash-search(T, K)

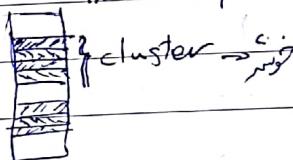
1. $i = \emptyset$
2. repeat
3. $j = h(K, i)$
4. if $T[j] == K$ then
5. return j
6. $i = i + 1$
7. until $T[j] = \text{null}$ or $i = m$
8. return null

($\log n$) uniform hashing

if $m!$ inserting $\frac{1}{m!}$ success probability probe value

probe sequence

linear probing



$$h(K, i) = (h(K) + i) \bmod m$$

quadratic probing

$$h(K, i) = (h(K) + C_1 + C_2 \cdot i^2) \bmod m$$

double hashing

$$h(K, i) = (h_1(K) + i \times h_2(K)) \bmod m$$

Subject

Date

طريق حل ملخص ديناميكية الموارم

(الجزء) Divide & Conquer ①

• طرق حل ملخص ديناميكية الموارم ①

• حل ملخص ديناميكية الموارم ②

• حل ملخص ديناميكية الموارم ③

Fact(n) ④

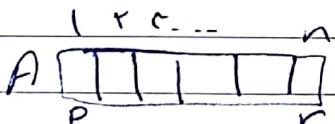
if $n=1$ then return 1

else return $\text{Fact}(n-1) * n$

• طرق حل ملخص ديناميكية الموارم ④

• طرق حل ملخص ديناميكية الموارم ④

$A \rightarrow n$ Max ④



$$k = \log(n) \leftarrow n = r^k$$

max(A, P, R)

$$nT(n) + \frac{n}{r} O_{c1} + \dots + O_1 = O(n)$$

1. if $P > R$ then return $A[P]$ } 0,

2. $Q \leftarrow \lfloor \frac{P+R}{r} \rfloor$

3. $m_i = \max(A, P, Q)$ $T(n/r)$

4. $m_r = \max(A, Q+1, R)$ $T(n/r)$

$$T(n) = r T(n/r) + O_{c1}$$

$$= r [r T(n/r) + O_{c1}] + O_{c1}$$

5. if $m_i > m_r$ then } 0,

ART

$$= r^k T(\frac{n}{r^k}) + r^{k-1} O_{c1} + \dots + O_1$$

6. else return m_i } $O(1)$
return m_r

Subject

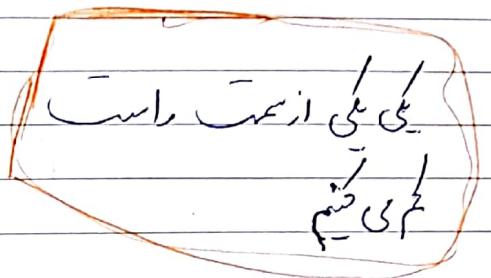
Date $\max(A, P, 8) \quad T(n)$

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

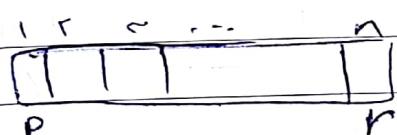
2. $m_1 = \max(A, P, R-1)$

3. if $m_1 > A[r]$ then return m_1

4. else return $A[r]$



$$T(n) = T(n-1) + O_1 \rightarrow T(n) = nO_1 = O(n)$$



Time complexity

Mergesort (A, P, R) {

1. if $P >= R$ return

$$T(n) = T(n/2) + \Theta(n)$$

2. $q \leftarrow \lfloor \frac{P+R}{2} \rfloor$

$$= n \log(n)$$

3. merge - sort (A, P, q) {

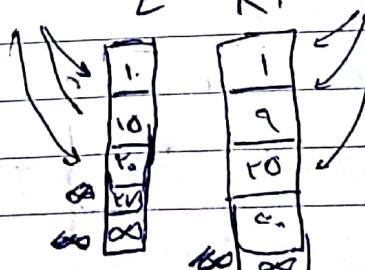
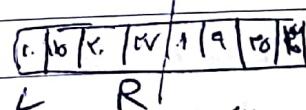
$$\text{Top half } T(n/2)$$

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

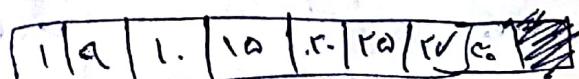
$$T(n/2)$$

5. merge (A, P, q, R) } $\Theta(n)$

merge (A, P, q, R) {



ART



Subject

Date

merge(A, P, Q, R)

$$O(n) = R(n) = \Theta(n)$$

C₁. $n_1 \leftarrow q - p + 1, n_r \leftarrow r - q$

C₂(n₁) 2. for $i \leftarrow 1$ to n_1 , do

لـ $L[i] \leftarrow A[P+i]$

C₃(n₁) 3. $L[n_1+1] \leftarrow \infty$

وـ $R[n_1+1] \leftarrow \infty$

C₄(n_r) 5. for $i \leftarrow 1$ to n_r do

C₅(n_r) 6. $R[i] \leftarrow A[q+i]$

C₆ 7. $R[n_r+1] \leftarrow \infty$

C₇ 8. $i \leftarrow 1, j \leftarrow 1$

C₈(n₁) 9. for $k \leftarrow p$ to r do

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

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

i++

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

j++

کوہی دیر سے اسی طرز میں ایسا کام کیا جائے کہ

1cc n-1 n

III □ Q key \approx اول (n-1), خانہ اسی طرز میں کیا جائے کہ

ART

92

Subject

Date

2.3 Insertion sort دیگر ساده‌ترین

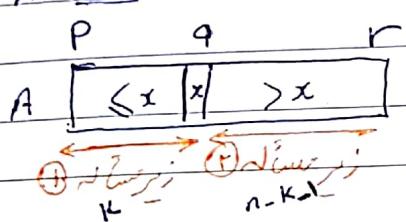
حالات ابتدا $O(n^2)$ بعده $O(n^2) = O(n^2)$

حالات ابتدا $O(n)$

Quick sort

III

حالات ابتدا



Quick-sort (A, P, r) {

1. if $P \geq r$ then return $O_{(1)}$

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

3. Quick-sort ($A, P, q-1$) $T(k)$ $O_{(k)}$

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

5. \rightarrow $O_{(1)}$

$$T_0 = C$$

$$\text{O.H. } T_n = T(k) + T(n-k-1) + O_{(1)} + O_{(n)}$$

که دری $< k < n-1$

$$\rightarrow T_{n-1} = T_1 + T_{n-1} + O_{(n)} = \boxed{O_{(n^2)}} + T_{n-1} + O_{(n)}$$

\downarrow
 $k=0$

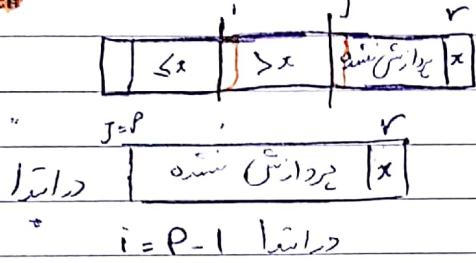
ART

وچیخی دهد که آرایه در دری از است امرتب شده نزولی باشد
خانه‌ی آفر عنصر \min و بقیه‌ی آرایه امرتب صعودی باشد

از این

Subject

Data



ماسٰ حلفر

اگر از "swap" بود
ساده تر

partition (A, P, r) {

α_{n+1}

دائرہ بزرگ سیاست
سکول ایکارنی ٹکنولوژی
جی کریسٹنی مائی

$$1. i = p-1, x = A[r]$$

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

3. if $A[j] \leq x$ then

4. litt

5. $\text{swap}(A[i], A[j])$

6. Swap ($A[i]$, $A[i+1]$) } ↗ جو فیکر می رہے تو اس کا

$\max_{1 \leq i \leq n} f(i) = k$ \iff $\exists i \in \{1, \dots, n\}$ $f(i) = k$

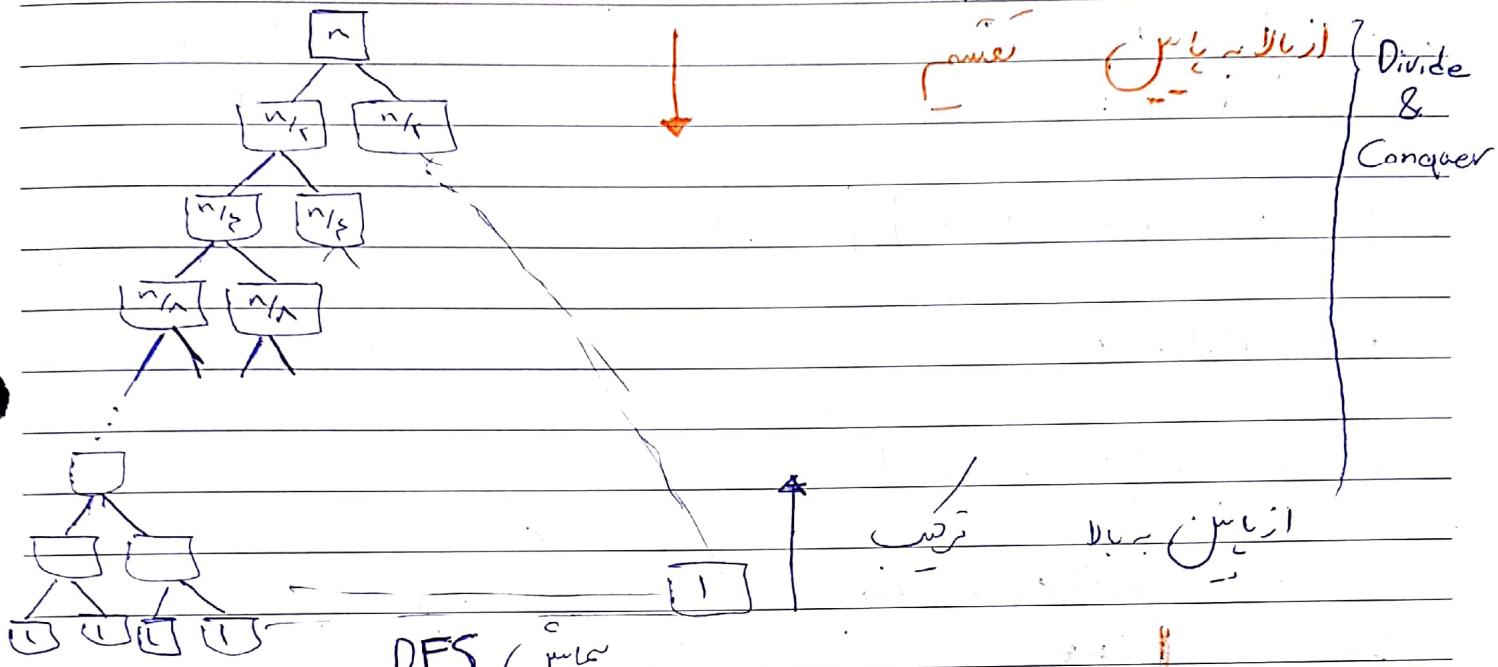
$$T(n) = T(n-1) + T(1) + O(n^k) = T(n-1) + O(n) = O(n^k)$$

$$T(n) = T\left(\frac{n}{r}\right) + T\left(\frac{n}{r} - 1\right) + O(n) \quad ; \quad k = \frac{n}{r}$$

$$= RT \left(\frac{n}{r} \right) + O(n) = O(n \log n)$$

Dynamic programming = D.P

روش برنامه ریزی پیوسته



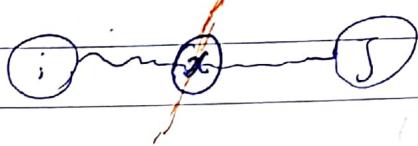
قطع از پاسن سالار سالار را حل می کند D.P روشن

۲- حافظه در درستگاه تکراری برای حل می کند

ساختاری دارد که درین دیرگی زیرساخت را برای داد (optimal substructure) D.P

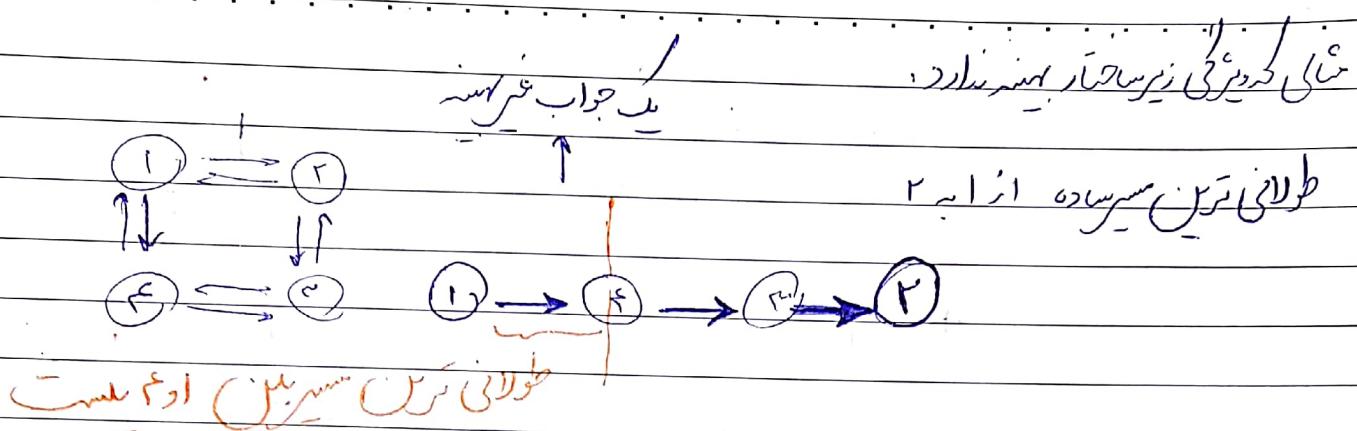
لهم بعین کوئد تمیت شری (ایم دهد فهم اینست ورد)

۳- از نیاز حسنه در دیرگی زیرساخت را برای داد



سکولی حیاتان می خراف در حرف

لهم حسن را و قیچی بی سلیم که هر دو حیاتان می خواهد



لرستانی نمای جواب داشته باشد و تحریب فرد باندی \rightarrow D.P

حل ساده برای تعدادی بین خوب و نایل

$$M_1 \times M_r \times M_s \times M_k \dots \times M_{n-1} \times M_n =$$

به دنال ترین تعداد خوب هستم

از خوبی اول همیهم و $n-1$ = خوب آخر

$$P_{cn1} = \sum_{k=1}^{n-1} p_{ck} \cdot P_{c(n-k)} = \frac{C^n}{n!} \cdot 1$$

$$P_{q,r} \times P_{q,r} = P_{q,r} \quad \text{تقریباً} = p \cdot q \cdot r$$

$$M_{p,q} \quad M_{q,r} \quad M_{p,r}$$

هر چیز

$$\text{خوب: } ((M_1 \times M_r \times M_s \times \dots \times M_k) \times (M_{k+1} \times \dots \times M_n))$$

جواب

بین خوب

Subject

Date

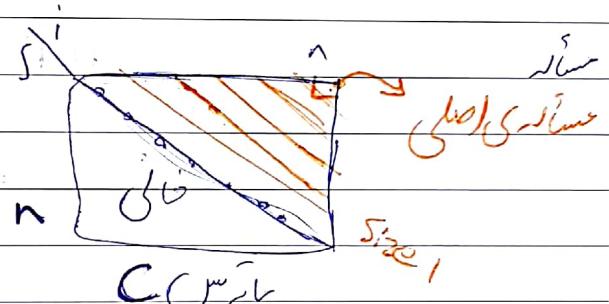
$$P_{i,j} = (M_i \times M_{i+1} \times \dots \times M_k) \times (M_{k+1} \times \dots \times M_j)$$

الى الـ i الـ $i+1$ الـ \dots الـ k الـ $k+1$ الـ \dots الـ j

$P_{i,j}$ \leftarrow تعداد ضروري $\leftarrow P_{i,k+1,j} + P_{i,k+1,j}$ \leftarrow تعداد ضروري $\leftarrow P_{i,k} + P_{i,k,j}$

$$C_{i,j} = \min_{i \leq k \leq j} (d_{i,i} \times d_k \times d_j + C_{k+1,j}) + C_{i,k}$$

$$P_{i,i}; C_{i,i} = 0$$



$$d_{i,i} d_{i,i} d_{i,i} d_{i,i} d_{i,i}$$

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

$\times \dots \times \dots \times \dots \times \dots$

$$(d_i^k)$$

$$C_{i,c} = \min_{1 \leq k \leq c} \{ C_{i,k} + C_{k+1,c} + d_0 d_k d_c \}$$

$$= C_{i,1} + C_{r,r} + d_0 d_r d_c = 0 + 0 + 1 \times 0 \times 1 = 0$$

$$C_{r,r} = C_{r,r} + C_{c,c} + d_0 d_r d_c = 0 + 0 + 0 \times 0 \times 0 = 0 \quad k=r$$

$$C_{c,c} = 0 + 0 + 0 \times 0 \times 0 = 0 \quad k=c$$

$$C_{i,r} = \min_{1 \leq k \leq r} \{ C_{i,k} + C_{k+1,c} + d_0 d_k d_c \}$$

ART

Subject

Date

$$= \min \begin{cases} K=1: C_{1,1} + C_{r,1} + d_1 d_r = 0 + 1 \dots + 1 \times 2 \times 1 = 11 \dots \\ K=r: C_{1,r} + C_{r,r} + d_r d_r = 0 \dots + \dots + 1 \times 1 \times r = V_{\dots} \end{cases}$$

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

$$\min \begin{cases} K=r: C_{r,r} + C_{r,f} + d_r d_r d_f = 0 + r \dots + 0 \times 1 \dots = V_{\dots} \\ K=f: C_{r,f} + C_{f,f} + d_f d_f d_f = 1 \dots + 0 + 2 \times r \times 1 = 11 \dots \end{cases}$$

$$C_{1,f} = \min \{ C_{1,K} + C_{K+1,f} + d_K d_K d_f \}$$

$$= \min \begin{cases} K=1 \Rightarrow C_{1,1} + C_{r,f} + d_1 d_1 d_f = 0 + 11 \dots + 1 \times 2 \times 1 = 19 \dots \\ K=r \Rightarrow C_{1,r} + C_{r,f} + d_r d_r d_f = 0 \dots + r \dots + 1 \times 1 \dots = V_{\dots} \end{cases}$$

$$K=f \Rightarrow C_{1,f} + C_{f,f} + d_f d_f d_f = 11 \dots + 0 + 1 \times r \times 1 = 19 \dots$$

i	1	r	c	f	
$0 \dots$	$0 \dots$	$11 \dots$	$1,1 \dots$		
$K=1$	$K=1$	$K=r$			\rightarrow $\text{SLI}_{1,1}$
$1 \dots$	$1 \dots$	$11 \dots$			
$K=r$	$K=r$	1			
$2 \dots$	$2 \dots$	$1 \dots$			
$3 \dots$	$3 \dots$	$1 \dots$			
\vdots	\vdots	\vdots			
$n \dots$	$n \dots$	$1 \dots$			

ART

(VI)

Subject

Date

$$\underbrace{(n+r)}_{\text{number of terms}} \times \underbrace{n^r}_{\text{last term}} = D.P$$

$DC(i,j)$

1 min = ∞ j-1 if (i == j) return 0

2. For $k=1$ to need do

$$m = DC(i, k) + DC(k+1, j) + d[i-1] \cdot d[k] \cdot d[j]$$

Q. if $m < n$ then $\min = m$

5. return min

10

~~apologize~~ is!

$$T(n) = \sum_{k=1}^{n-1} T(k) + T(n-k) + Q_i \stackrel{\text{recursion}}{=} \sum_{k=1}^n T(k) + Q_i$$

optimal substructure!

$$\Theta(n^k) = O(c^n) \rightarrow$$

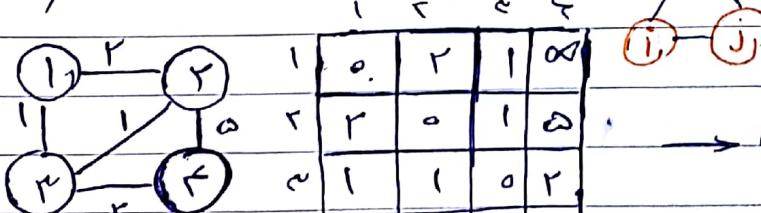
مکالمہ میں اپنے بھائی کو کہا جائے۔

Gree dy

-t - lew T (j, v) H

Greedy choice is wrong

Floyd-warshall (all pairs)



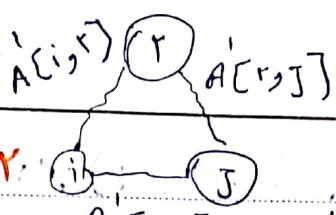
اللّهُمَّ إِنِّي سَأْتَكُنْ لِأَنْزَافِ

- 3 -

مکانیزم انتشار

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

$A^r = 1$	0	r	1	V
r	0	1	1	a
1	1	1	1	r
r	V	a	r	0

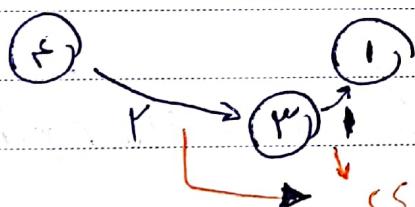


$$A^K[i, j] = \min\{A^{K-1}[i, j], A^{K-1}[i, k] + A^{K-1}[k, j]\}$$

 $K = r$

$A^r =$	0	r	1	r
r	0	1	1	r
1	1	0	r	
r	r	r	r	0

$A^r =$	0	r	1	r
r	0	1	1	r
1	1	0	r	
r	r	r	r	0



جواب نیز است A^n

کوشاں اس کا جواب A^n ہے

ذیری خود راست باندھ

جنریشن: $\text{for } n$

($A_{ij}Q_{ij}$) \rightarrow خانہ کا سے بڑی جیتے وہ عرض کا $O(n^2)$ دھریں

$$n \times O(n^r) \times O(1) = O(n^r) \quad \text{لما} \quad n \times O(n^r) = O(n^r)$$

$$\Theta(n^r) \rightarrow \omega^r_3 = \Theta(|V|^r)$$

اون (الورثم داسیت) P. P

Dijkstran $\leq O(|E| \cdot \log |V|)$ بديهيا

Greedy

البرنس حسام

نَخْبَرْهُمْ : سَلَّمٌ خَرْدَرْلَنْ (سَلَامٌ) لِـ خَرْدَرْلَنْ (سَلامٌ) سَعَادَ سَارَ (سَلامٌ) سَعَادَ سَارَ

انجیب حرصان ہے اور دست مانچی سکھ لئے کی بزرگ تھا۔

شیل (شیل) اسکندر طا طباون، سکنہ ۵ و ۳ و ۱

$$(\Delta = \Gamma_x \Delta = \text{curl} \text{ curl})$$

$$C_1 = \omega_0 r_1 + k_1 x_1 \rightarrow 1, r_1, \omega_0 \text{ and } C_1 \text{ are given}$$

$$C_0 = \tau^{-1} \tau_0 - \tau_0$$

مکاریں بنی کتاب (Task), یعنی درایزن میں برداشت

Task, in (j). اس فیلٹر میں بعدی نیوڈز نہ اچھا سوں۔ اس تریکھ میں



مشخصه $T_1 = t_1$

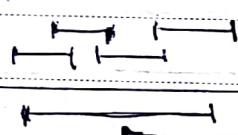
برای اجرای برودری $\langle T_1, T_F, T_C \rangle$

$$T_F = t_1 + t_r \quad T_p = t_1 + t_r + t_{pF}$$

$$\text{زمان نیاز دارد} = \frac{3t_1 + 2t_r + t_{pF}}{2} \rightarrow \text{زمان نیاز دارد} = \frac{\text{زمان نیاز دارد}}{2}$$

انتخاب حریف: کارخانه را برای تهییب صوری زیال برداشت آن های ریس کن.

$A_i: (S_i - f_i)$ Activity selection سل) انتخاب فعالیت



انتخاب حریف: طلاس میام

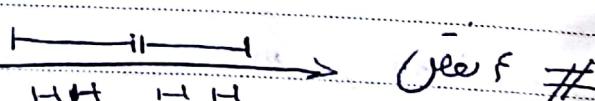
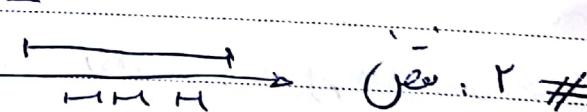
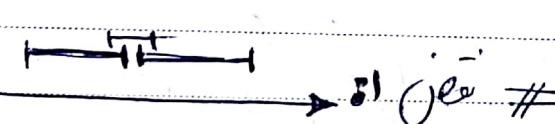
انتخاب حریف ①: اولویت فعالیت های کمتر میان.

انتخاب حریف ②: اولویت فعالیتی که تکمیل شده باشد. (فعالیتی که نیافروده است)

انتخاب حریف ③: اولویت فعالیتی که تکمیل شده باشد و بقید را درد.

انتخاب حریف ④: اولویت انتخاب ۲ تایی های تغیراتی با محض دارند.

$f_i: @$



Subject

Date

$$s^* \xrightarrow{A^*} f^*$$

این ایجاد شود:

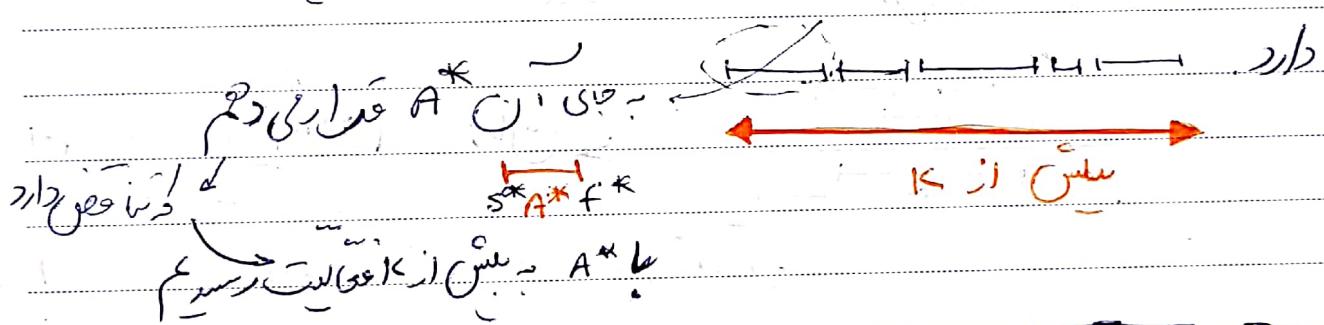
$$A^* = (s^*, f^*)$$

یا فرض اینکه A^* غرض جدید است که این حراب

را پس از این وابح حراب که عوایت دارد.

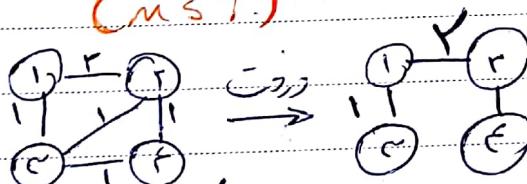
لذا: همچو حراب دستی در مسائل A^* نباشد و بین از K عوایت

این ایجاد شوند: \leftarrow ایجاد خود را در A^* در آن سمت و بین از K عوایت



Minimum Spanning Tree

(MST)



$$w = 4 = 2 + 1 + 1 = 4$$

مسئله بزرگ: دست یافتن

الgoritم چون: یک هاده را در نظر بگیر و صورت مرتبی شوند و بین

تریک انتخابی خصم اخراج کرد و یک یک باعث اید و در بدل قدری عیلاً آنکه

نهی شود این بدل ما انتخاب غیر خصم است. الgoritم Kruskal

P4PCO

$$O(E) \times DFS \quad |V| = E \cdot V$$

$$(|V| + E)$$

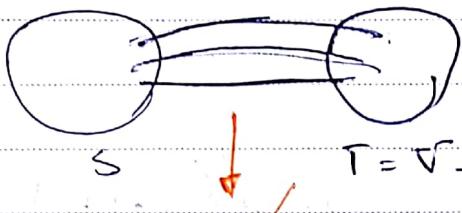
Subject _____

Date _____

$$V = S \cup T \quad S \cap T = \emptyset$$

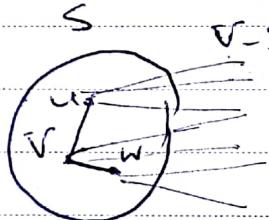
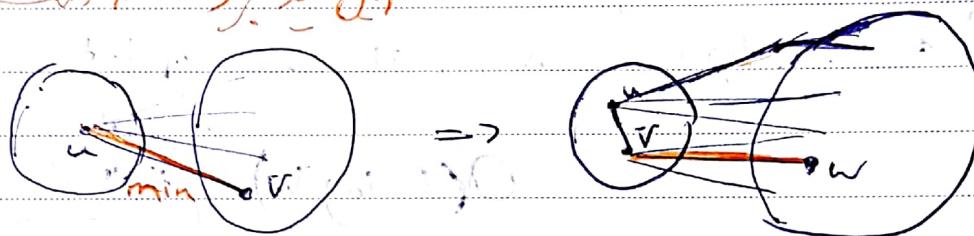
دسته دوست نیزی داشت

آخر



$$S \quad T = V - S$$

MST



$$S$$

$$V-S$$

$$V-S$$

درین الگوریتم در راهی کوچک خواهد بود. (آخر درین عالم)

امحال وجود یک پسندیده ازینها ازین یورون

O

: Prim // الگوریتم

مینینین درین عالم قدرسته. (برین رانی کوچک و می تین طور تکاری داشت)

Kruskal // الگوریتم

sort

انتخاب های سریع

O(E.V)

$E \times \log(E)$

درجهات کنکل شدن

برای محقق کار

DFS

برای بیان گردش

$\Theta(E \log(V))$

|E|, |V|

حذف در جمله ایجاد

کسر

PAPCO

VV

Subject

Date

اگر جی اے DFS میں label برائی دوئے تو سارے label اےDFS میں

$$V-1 \times O(V) = O(V)$$

فریان جریں
کے ذریعہ

اگر جو راہ وحیت خود برائی دوئے label

$O(\log V, V) \times \left\langle \left\langle \log(V) \right\rangle \right\rangle \rightarrow \text{اے، اے، اے} \leftarrow \text{update}$ برائی کے لئے

$O(E \log V)$ فریان طل اللوریم

set operations i ویسے

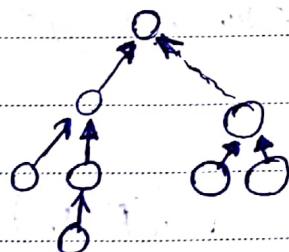
Find(S, x) Del(S, x)

Add(S, x)

union(S_i, S_j)

جی کی کی کی

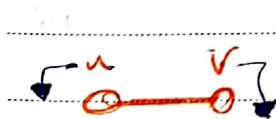
Rooted Tree



Find $O(\log V)$

union $O(1)$ → point 1، 2، 3، 4، 5

Add, Delete $\rightarrow O(\log V)$ → جوں جوں



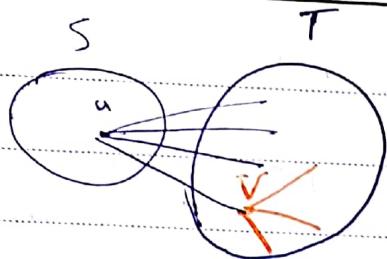
Add، جوں جوں

کھلے کریں

اگر جوں جوں نہیں کے دوست جوں جوں، اب بچھوئے اور کیا

PAPCO

جوں جوں $E \log(V)$



Prim

اگر ریشم

لے، لیست میں ہی تکمیل میں ہے

آن اختیاب شود

قاتلان: پر رائیں رفیع۔ دل درمی سرد دل ہی جائز کے ان وصل بروں نے خوف سے

دلی دیر دل ہی اون بے اضافہ می سوو

$$L_{\text{min}} = O(E) \leftarrow L_{\text{min}} (\text{سیناریو})$$

$$\textcircled{I} \quad O(E) = \sum_{w \in V} \text{degree}(w) \times O(1) : L_{\text{min}} \text{ (سیکھیں)} \downarrow \text{حروف اضافہ}$$

$V - 1$ سیناریو، \min (سیناریو)

(کے) : $O(V \cdot E)$

$$\textcircled{I} \quad \sum \text{degree}(w) \times O(\log E) =$$

Heap، جنہیں حروف اضافہ کے

post level minHeap از

$$E \cdot \log E = O(E \log V)$$

Subject

Date

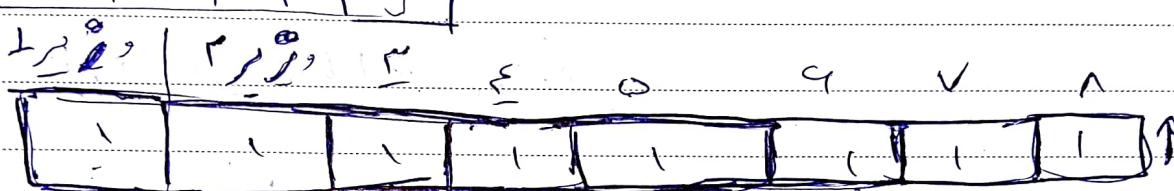
روش حسب ذکری فضای حالات:

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	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

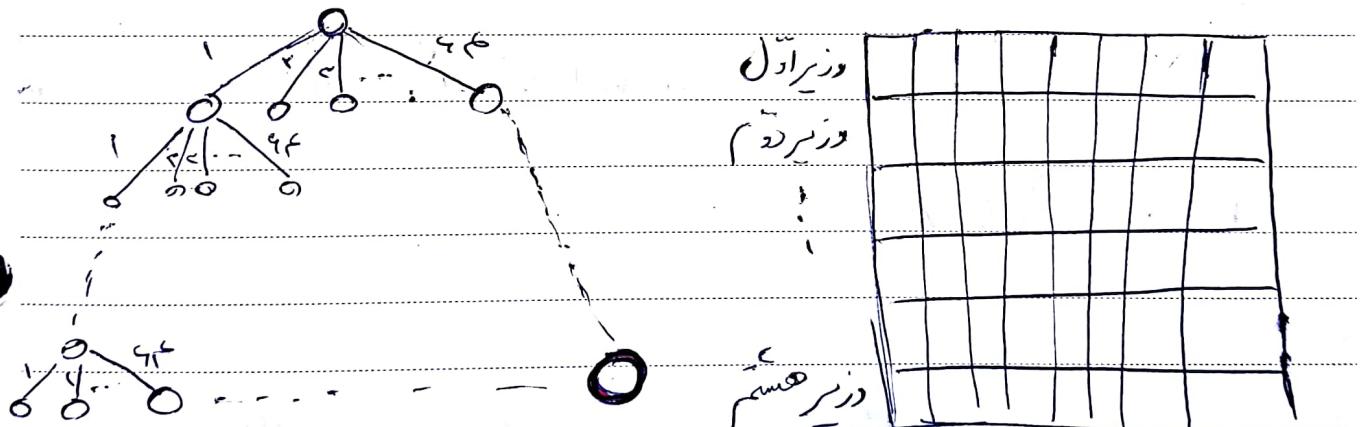
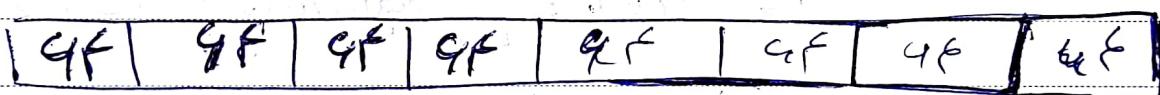
ساختی وزیر: \rightarrow فضای سطح اور پردازش

بنا بر داده های اینجا

\rightarrow $\text{وزیر} = \text{QF}$ $\rightarrow \text{پل} = (\text{QF})^*$ $\rightarrow \text{پل}(\text{حالات})$



پل کاسیده میگیرد و در پردازش



DFS بیانی

خطای سفر حالات میگیرد

(QF)

حال بستون (هاست تجربی گفته): A

$$\begin{array}{|c|c|} \hline & n! \text{co}(n!) \\ \hline & n! \text{cw}(n!) \\ \hline \end{array}$$

RAPCO

haeri@aut.ac.ir

branch & bound (فرز)

End-hard

(نحوه)

fractional

مسئله کسری

w_i مقدار میزان کسری و v_i مقدار میزان کسری ارزش

شرط: $\sum w_i \leq M$ برای اساید داخل گوله.

راحیاب کشم وزیری ها استفاده کنم. $\frac{v_i}{w_i}$ ارزش میزان کسری این طرداعم

Fractional

حال: بسیار اجازه نیست که از هر چند که را در این قطعه از طبله برداشت برداشته

این سه دلخواه

0	0	-1	1
---	---	----	---

حالت و جی تقاضا: ۳ⁿ حالت طبق

for i=0 to n-1

برنامه فرزنده (فرز) $M=14$

فرز	وزن	وزن	وزن	وزن
1	4.5	2	1	
2	2.5	0	2	
3	0.5	1.0	2	4

R4PCO

A1

تریک انتخاب سایر حساب

Subject _____
Date _____

-1 شرط ≤ 110 = fractional int

upper bound

