

Subject :

Date .....

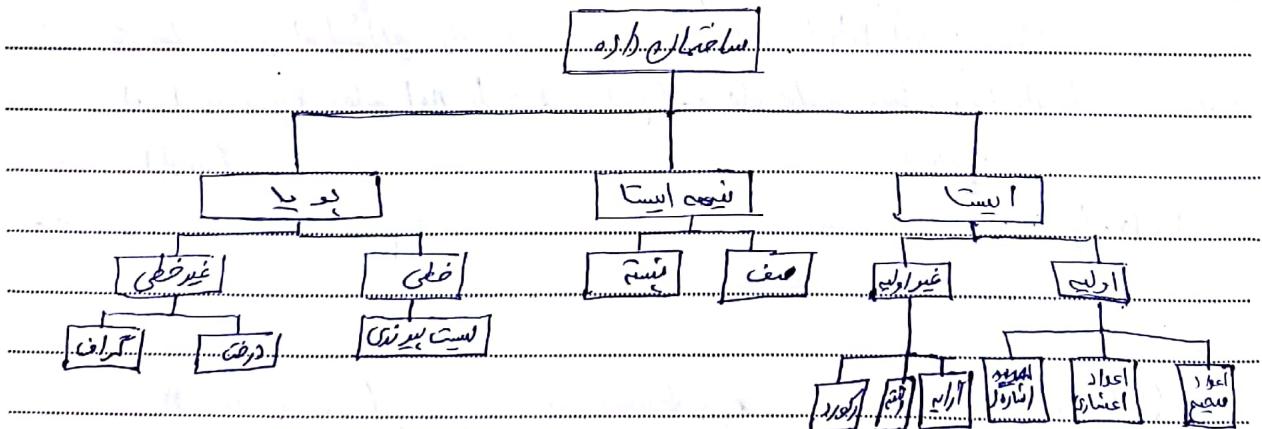
بر الأمانة: (دراز) الكورس (دراز) (النوع درجة) (دراز)  
دراز ساخن (دراز) درجة (النوع درجة) (دراز) (دراز)  
دراز (دراز) (دراز) (دراز) (دراز) (دراز) (دراز) (دراز) (دراز)

مراجع: موسى دارو، ساخن (دراز) (دراز)

ساخن (دراز) (دراز) (دراز)

الكورس: موسى دارو، ساخن (دراز) (دراز) (دراز)  
دراز (دراز) (دراز) (دراز) (دراز) (دراز) (دراز) (دراز)  
دراز (دراز) (دراز) (دراز) (دراز) (دراز) (دراز) (دراز)  
الكورس: موسى دارو، ساخن (دراز) (دراز) (دراز)

ساخن (دراز) (دراز) (دراز) (دراز) (دراز) (دراز) (دراز) (دراز)  
دراز (دراز) (دراز) (دراز) (دراز) (دراز) (دراز) (دراز)



Subject :

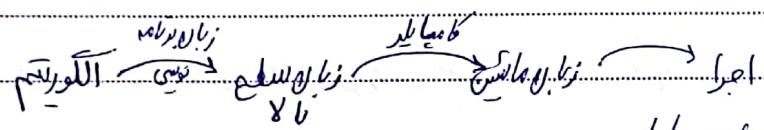
Date .....

العنصر الأول دليل يدل على العنصر الثاني كونه ملحوظ أي كباقي العناصر في المجموعة التي لا ينتمي لها العنصر الثاني

نحو عاليات العدد ونحو صور ونحو أشياء معرفة ونحو أفعال المعرفة

نحو خصائص الگوریتمها: ملحوظ، معرف، معلوم، اجراء المعرف

نحو بعدهم الگوریتم از تکرار هر زمان اجراء دو روکش اندازه گیری و نتایج را در آینم



دو روکش اندازه گیری دو الگوریتم تمام گذشتند و نتایج

برآورده شدند لایم (Lime) (ستون زمانی) و همچنان شمعون (Shmuon) (نوع زمانی) برآورد شدند

نوع سیمان و شمعون را که اندیشیدند (Lime and Shmuon)

function:  $f(1 \{ \rightarrow \} )$

int a  $\rightarrow 1$

$a = 0$

for ( $i=1$ ;  $i < n$ ;  $i++$ )

$a++$

return a

نحو عالیات و مطابق زمان اجراء اینجا مذکور شد (1)

( $x = 0$  (از زمان اجراء اینجا مذکور شد)  $\rightarrow$  (زنگنه اینجا مذکور شد)) (1)

نحو عالیات زمان اجراء اینجا مذکور شد زمان اجراء (2)

(من لایم اینجا برای for زمان اندیشید (Lime)  $\rightarrow$  باید عالیات زمان اجراء برای اینجا باشد) باعتراد

return

نحو عالیات (2)

$1 + (n+1) + n + 1$

نحو عالیات عکس که باز طریم

$x = 0$

for ( $i=1$ ;  $i < n$ ;  $i++$ )

$$\sum_{i=1}^n \sum_{j=1}^n 1 = \sum_{i=1}^n n = n^2$$

for ( $j=1$ ;  $j < n$ ;  $j++$ )

MICRO  $\frac{x++}{\text{return } x}$

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

Subject :

Date .....

$$\begin{aligned}
 & x_{\text{for}} = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n 1 = n^3 \\
 & \text{for}(i=1; i \leq n; i++) \\
 & \quad \text{for}(j=1; j \leq n; j++) \quad \text{with } O(j) \text{ per loop} \rightarrow \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n 1 = n \cdot \frac{n(n+1)}{2} \\
 & \quad \quad \text{for}(k=1; k \leq n; k++) \\
 & \quad \quad \quad \text{with } O(1) \text{ per loop} \rightarrow \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n 1 = \sum_{i=1}^n \sum_{j=1}^n j \\
 & \quad \quad \quad = \sum_{i=1}^n \frac{i(i+1)}{2} = \frac{1}{2} \left[ \sum_{i=1}^n i^2 + \sum_{i=1}^n i \right] = \frac{1}{2} \left( \frac{n(n+1)(2n+1)}{6} \right) \left( \frac{n(n+1)}{2} \right)
 \end{aligned}$$

$$\begin{aligned}
 & x = 1, b = 1 \quad n \quad 1 \\
 & \text{while}(i \leq n) \quad \frac{1}{\sqrt{b}} \quad \frac{1}{\sqrt{b}} \quad \sqrt{b} \leq n \rightarrow k \leq \log n \rightarrow k \leq \log n + 1 \\
 & \quad \quad \quad i = i/\sqrt{b} \quad \frac{1}{\sqrt{b}} \quad \frac{1}{\sqrt{b}} \\
 & \quad \quad \quad \vdots \quad \vdots \quad \downarrow \text{with a gap of size } \sqrt{b} \\
 & \text{return } n; \quad k \quad \frac{1}{\sqrt{b}} \leq n
 \end{aligned}$$

$$\begin{aligned}
 & x = 1, i = n \\
 & \text{while}(i \geq 1) \quad n \quad 1 \\
 & \quad \quad \quad i \quad \frac{1}{n} \quad \frac{n}{\sqrt{k-1}} \leq 1 \rightarrow k \geq \log n + 1 \\
 & \quad \quad \quad i = i/\sqrt{b} \quad \frac{1}{n} \quad \frac{n}{\sqrt{k-1}} \\
 & \quad \quad \quad \vdots \quad \vdots \\
 & \text{return } n \quad k \quad \frac{n}{\sqrt{k-1}}
 \end{aligned}$$

$$\begin{aligned}
 & x = 0 \\
 & \text{for}(i=1; i \leq n; i++) \quad i \quad x \\
 & \quad \quad \quad i \quad \frac{1}{\sqrt{r}} \\
 & \quad \quad \quad j = 1 \quad \frac{1}{\sqrt{r}} \quad \frac{n}{\sqrt{r}} \\
 & \quad \quad \quad \text{while}(j \leq i) \quad \frac{1}{\sqrt{r}} \quad \frac{1}{\sqrt{r}} : \quad \sum_{l=1}^n \lceil \log(i+1) \rceil \\
 & \quad \quad \quad j = j * r \quad n \quad \log n + 1 \\
 & \quad \quad \quad \vdots \quad \vdots \\
 & \quad \quad \quad \text{ref. U.C. } x
 \end{aligned}$$

### **Subject :**

Date .....

$j \rightarrow \frac{n}{r} \leftarrow n$	$i$	$x$
1	1	$\frac{n}{r}$
$j \rightarrow \frac{n}{r} \leftarrow \frac{n}{r}$	$r$	$\frac{n}{r}$
1	$r^2$	$\frac{n}{r^2}$
1	$r^3$	$\frac{n}{r^3}$
	$\vdots$	$\vdots$
1	$r^k$	$\frac{n}{r^k}$

return  $x$

Ques:  $n \left( \underbrace{\frac{1}{r} + \frac{1}{r^2} + \frac{1}{r^3} + \dots}_{k \text{ terms}} \right) = n$

$x = 0$	$i$	$x$
for ( $i=1$ ; $i \leq n$ ; $i++$ )	$i$	$n$
	$p$	$n-p$
for ( $j=1$ ; $j \leq n$ ; $j++$ )	$j$	$n-p$
	$p$	$n-p$
$x = 0$	$\frac{n}{p}$	$n - (\frac{n}{p} - 1)$
$n =$	$\frac{n-1}{p}$	
return $x$	$= \frac{n^p}{p} - \sum_{i=0}^{n-1} i \cdot \frac{n^p}{p} - \frac{(\frac{n-1}{p}) \frac{n}{p}}{p}$	

insertion sort ( $A, n$ )

for(k:=2 to n) → n

key = A[k] → n-1

$$i = k - 1 \dots n - 1$$

while(*i* > 0 and  $A[i] > key$ ) {

$$f(n) = n + (n-1) + (n-1) + \sum_{k=p}^m t_k$$

$$A[i+1] = A[i]$$

$i_m, i-1$

$A[i+1] = \text{key}$

三

بجزئي مالك:  $t_k = 1$  ← وروجت بضرر ملوك ملكي ملك  
بجزئي مالك:  $t_k = M$  ← وروجت بضرر تجعل ملكي ملكي ملك

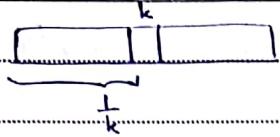
$$\frac{1}{n!} \sum_{i=1}^n \sum_{k=p}^n t_k = \sum_{k=p}^n \underbrace{\sum_{i=1}^n \frac{t_k}{n!}}_{t_k}$$

MICRO

Cours Capitaliste

Subject :

Date .....

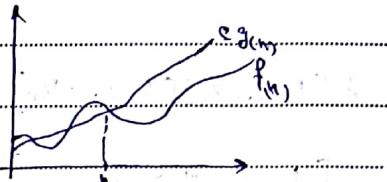


$$T_k = \sum_{j=1}^k \frac{1}{k} (k \cdot j + 1)$$

 $\theta(n) = \Theta(n)$ 

big o : O

$$c_1, n_0 > 0 \quad f(n) = O(g(n)) \iff n \geq n_0 \implies f(n) \leq c_1 g(n)$$



$$\text{def: } f(n) = kn^r + rn \in O(n^r) \quad kn^r + rn \leq cn^r \quad n \geq n_0$$

$$kn^r + rn \leq cn^r \quad c = r, n_0 = r$$

... big o (عکس کارهای پیشنهادی) ...

$$f(n) \in O(x^m) \leftarrow \text{big o (عکس کارهای پیشنهادی)} \quad f(n) = a_m x^m + a_{m-1} x^{m-1} + \dots + a_0$$

... امثالی:  $O(1) < O(\log n) < O(n) < O(n \log n) < O(n^r) < O(r^n) < O(n!) < O(n^r)$  ...

big : Ω

$$c_1, n_0 > 0 \quad f(n) \in \Omega(g(n)) \iff n \geq n_0 \implies c_1 g(n) \leq f(n)$$

$$\text{def: } f(n) = \Delta + n^r \in \Omega(n^r) \quad \Delta + n^r \geq c n^r$$

$$f(n) \in \Omega(x^m) \leftarrow \text{اماکنی (عکس کارهای پیشنهادی)} \quad f(n) = a_m x^m + a_{m-1} x^{m-1} + \dots + a_0$$

$$\theta: c_1, c_2, n_0 > 0 \quad f(n) \in \theta(g(n)) \iff c_1 g(n) \leq f(n) \leq c_2 g(n)$$

$$\text{def: } f(n) = \frac{1}{\gamma} n^r \quad f(n) \in \theta(g(n))$$

$$c_1 n^r \leq \frac{1}{\gamma} n^r - \gamma n \leq c_2 n^r \quad c_1 n^r \leq \frac{1}{\gamma} n^r - \gamma n \rightarrow c_1 \leq \frac{1}{\gamma} - \frac{\gamma}{n} \rightarrow n_0 \geq V$$

$$\frac{1}{\gamma} n^r - \gamma n \leq c_2 n^r \rightarrow c_2 = \frac{1}{\gamma} \rightarrow n_0 = \infty \rightarrow V = \infty, c_1 = \frac{1}{\gamma}, c_2 = \frac{1}{\gamma}$$

Subject :

Date .....

$A[L_1 \dots L_r][U_1 \dots U_r]$  as type  $\text{matrix}[U - L_r + 1] \times N$

$$\text{Index} = (i-1) \times n + j \quad \text{for } i=1 \dots U_r, j=1 \dots L_r$$

$A[L_1 \dots L_r][U_1 \dots U_r][L_p \dots L_q]$   $\text{index} = (U_r - L_r + 1)(U_q - L_q + 1)$

$$\begin{bmatrix} 1 & P & P & F \\ 2 & 4 & V & A \\ 9 & 10 & 11 & 12 \end{bmatrix} \rightarrow \begin{array}{|c|c|c|c|c|} \hline & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 1 & 0 & 1 & 0 & 1 \\ \hline 2 & 0 & 1 & 0 & 1 & 0 \\ \hline 3 & 1 & 0 & 1 & 0 & 1 \\ \hline \end{array}$$

After shifting indices

$A[i \dots j][l \dots m]$

$$\text{Index} = [(i - L_r) \times (U_r - L_r + 1) + (j - L_r)] \times n + l$$

$$\text{Index} = [(j - L_r) \times (U_r - L_r + 1) + (i - L_r)] \times n + l$$

$A[L_1 \dots U_r][L_p \dots U_r]$

$A[L_1 \dots U_r][L_p \dots U_r][L_q \dots U_q]$

$$\text{Index} = A[i \dots j \dots k] \rightarrow [(i - L_r) \times (U_r - L_r + 1) \times (U_r - L_r + 1) + (j - L_r) \times (U_r - L_r + 1) + (k - L_r)] \times n + l$$

$$\text{Index} = A[i \dots j \dots k] \rightarrow [(k - L_r) \times (U_r - L_r + 1) \times (U_r - L_r + 1) + (j - L_r) \times (U_r - L_r + 1) + (i - L_r)] \times n + l$$

Now calculate  $m \times n \times k$   $\leftarrow$   $A_{m \times n} \times B_{n \times k}$

$m \times n \times k$   $\leftarrow$   $A_{m \times n} \times B_{n \times k}$

**Subject :**

Date .....

$\vdash \forall x \forall y \forall z ((A_{min} \wedge B_{max}) \rightarrow C_{kj})$  (S. undas (S. für f.); S. (S. L. undas), L. 2)

وَالْمُؤْمِنُونَ الْمُؤْمِنُونَ (١٤) وَالْمُؤْمِنُونَ (١٥) وَالْمُؤْمِنُونَ (١٦)

$$\begin{array}{c} \text{Initial state: } \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \xrightarrow{\quad} \begin{bmatrix} F & F & F \\ 1 & F & F \\ F & F & F \\ F & 1 & 0 \end{bmatrix} \\ \text{Final state: } \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \xrightarrow{\quad} \begin{bmatrix} F & F & F \\ 1 & F & 0 \\ F & F & F \\ F & 1 & 0 \end{bmatrix} \end{array}$$

$\begin{bmatrix} e & y & f \\ d & y & v \end{bmatrix}$

LIFO  $\rightarrow$  Last In, First Out

push(item)      pop(item)

$\Sigma$  if ( $\text{top} == \text{maxStack}_-1$ ) { if ( $\text{top} == \text{maxStack}_-1$ ) .....

stack full      j.item( $\downarrow$ ) $\leftarrow$  top      stack empty

else.....newstack, if g is ld stack.....else.....(.....)

top = top + 1; int(s[i]) <= 1, item.push(i); return stack[top];

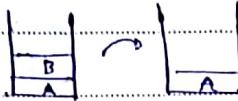
5 stack[top] = item

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

Subject :

Date .....

$$\begin{array}{c} A \\ \sqcup \\ B \\ \sqcup \\ S \end{array}$$



$$\begin{array}{c} B \\ \sqcup \\ S \end{array}$$

$$\begin{array}{c} C \\ \sqcup \\ S \end{array}$$

$$\begin{array}{c} S \\ \sqcup \\ S \end{array}$$

۱) خارج اون توابع، باید سیم کردن

۲) ارزیابی عبارت (P)

$A + B \leftarrow \text{infix}_4$  روش مانند و قوی

$AB + \leftarrow \text{postfix}_4$  روش لسو ندی

$+ AB \leftarrow \text{prefix}_4$  روش بیکنندی

$$(A+B) \times C \rightarrow AB + CX$$

$$axb+cx-ad \rightarrow ((axb)+c)-(a/d) \rightarrow ab \times c + ad /$$

ابعاد گشتن

خوب این این از پیشنهادی

خود قرار گشتر

$$\xrightarrow{\text{پیشنهاد}} - \rightarrow ab.c / ad$$

۳) ای

Email : Amir.Mohammad.Nazari.karbar@gmail.com

امیر محمد نظری

الgoritم دارای دو مرحله است، اولیه مرحله انتخابی (Selection Sort)، دوم مرحله انتخابی (Bubble Sort).

$A[n]$

for ( $i = 0$  to  $n-1$ )

    for ( $j = n-1$  to  $i$ )

        if ( $A[i] < A[t]$ )

$t < j$

            swap ( $A[i], A[t]$ )

کامپیوشن دارای دو مرحله است، اولیه مرحله انتخابی (Selection Sort)، دوم مرحله انتخابی (Bubble Sort).

از این دو مرحله انتخابی نیز  $n-1$  مرحله انتخابی دارد.

برای این دو مرحله انتخابی کامپیوشن دارای دو مرحله انتخابی دارد.

و در خاتمه این دو مرحله انتخابی دارای دو مرحله انتخابی دارد.

۱) ای

MICRO

Subject :

Date .....

for i = 1 to n

do for j = i+1 to n

do for k = i+1 to j do same o(1) statements

worst case?

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^n \sum_{k=1}^d 1 = \sum_{i=1}^{n-1} \frac{n(n+1)}{2} \cdot \frac{i(i+1)}{2} \cdot \frac{n(n+1)(n-1)}{6} = \frac{(n-1)n}{2} \cdot \frac{(n-1)n}{2} \cdot \frac{(n-1)n}{6} = O(n^3)$$

الحالات الممكنة  
أمثلة على الحالات الممكنة  
أمثلة على الحالات الممكنة  
أمثلة على الحالات الممكنة  
أمثلة على الحالات الممكنة

أمثلة على الحالات الممكنة  
أمثلة على الحالات الممكنة  
أمثلة على الحالات الممكنة  
أمثلة على الحالات الممكنة

for i = 1 to n

do if odd(i)

then for j = i to n do  $x_{i+j}$ else for j = 1 to i do  $y_{i+j}$ 

worst case?

$$\sum_{i=1}^n \sum_{j=p_{i-1}}^n 1 + \sum_{i=1}^n \sum_{j=1}^{p_i} 1 = \sum_{i=1}^n (n-p_i) + p_i = \sum_{i=1}^n n = \frac{n^2}{2} = O(n^2)$$

الحالات الممكنة  
أمثلة على الحالات الممكنة  
أمثلة على الحالات الممكنة  
أمثلة على الحالات الممكنة

Subject :

Date .....

بعض تعداد (نوع) از کارهای کمی کاری که در آن می‌باشد

$$\lg(\lg^{\alpha} n) \quad \lg^{*\alpha} n \quad (\lg n)^{\lg n} \quad n^{\lg n} \quad n! \quad (\lg n)! \quad \left(\frac{n}{\lg n}\right)^n \quad n^{\lg n} \quad (\lg n)^n$$

$$\lg(n!) \quad n^{\lg n} \quad \frac{1}{n^{\lg n}} \quad \ln \ln n \quad \lg^{*\alpha} n \quad \lg^{*\alpha} n \quad n^{\lg \lg n} \quad \ln n \quad 1$$

$$n^{\lg n} \quad (\lg n)^{\lg n} \quad e^n \quad \lg^n \quad (n+1)! \quad \sqrt{\lg n} \quad \lg^*(\lg n) \quad \sqrt{\lg n}$$

$$n \quad n^n \quad n \lg n \quad n^{n+1}$$

$$\frac{1}{n^{\lg n}} \quad 1 \quad \lg(\lg^{\alpha} n) \quad \lg^*(\lg n) \quad \lg^{\alpha} n \quad \ln \ln n \quad \sqrt{\lg n} \quad \lg n$$

$$(\lg n)^n \quad (\lg n)^{\lg n} \quad \frac{n}{\lg n} \quad \frac{n \lg n}{n!} \quad \frac{\lg n}{n^n} \quad n^n \quad (\lg n)! \quad \frac{(\lg n)^{\lg n}}{n^{\lg \lg n}}$$

$$\left(\frac{n}{\lg n}\right)^n \quad n^n \quad n \lg n \quad e^n \quad n! \quad (n+1)! \quad n^{n+1} \quad n^{n+1}$$

با شروع از دستگاه اول آغاز و هر دفعه بین این دستگاهها بازگشت (آغاز و بازگشت) insertion sort

با شروع از دستگاه اول هر دفعه کوچکتر از آن دستگاه را در دستگاه بعد از آن قرار (در دستگاه اول از دستگاه اول کوچکتر از آن دستگاه را در دستگاه اول قرار)

با شروع از دستگاه اول و هر دفعه کوچکتر از آن دستگاه را در دستگاه اول قرار (و بازگشت) bubble sort

کوچکتر از دستگاه اول دستگاه اول را در دستگاه اول قرار

infix, prefix, postfix, (عکس) infix, prefix, postfix, (عکس)

خروجی (output) نویسی و عملکرد (operation) stack (دستگاه) و خروجی (output) کوچکتر از دستگاه اول (آغاز و بازگشت) post-fix & infix

Subject :

Date .....

نحوه ترتیبی اولویت بالاتری از آن داشته باشد

مراجع

۱) مکانیزم خروجی

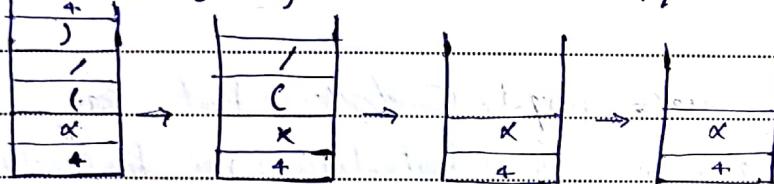
۲) پیکر و اتفاقی در لسخ اگر اولویت بالاتری از top است (غیر از اینجا) پیکر با اولویت بالاترین تن حذف و خروجی می شود و اینجا می خواهد پیکر با اولویت بالاترین تن حذف و خروجی می شود و اینجا می خواهد

۳) پرانتز باز + به لحاظ اتفاقی می شود

۴) پرانتز بسته : به لحاظ اتفاقی می شود اما اینجا پرانتز باز و بسته می خواهد

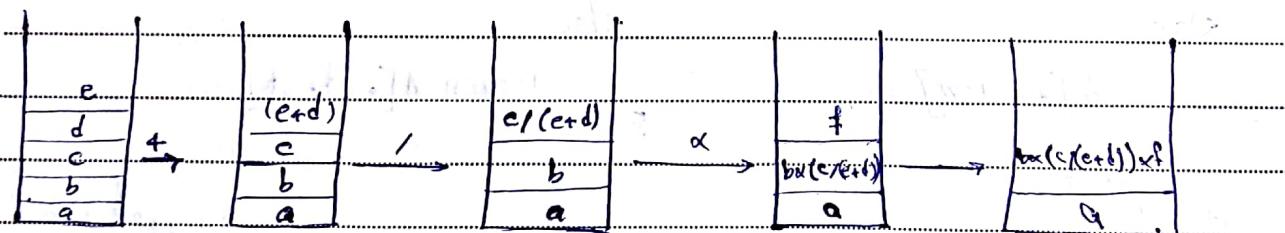
۵) پس از اینجا

$$a + b \alpha (c / (d + e)) \times f \rightarrow a b c d e + / \times f \times +$$



Q12

abced + x f x +



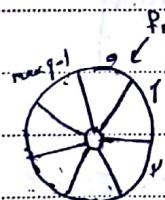
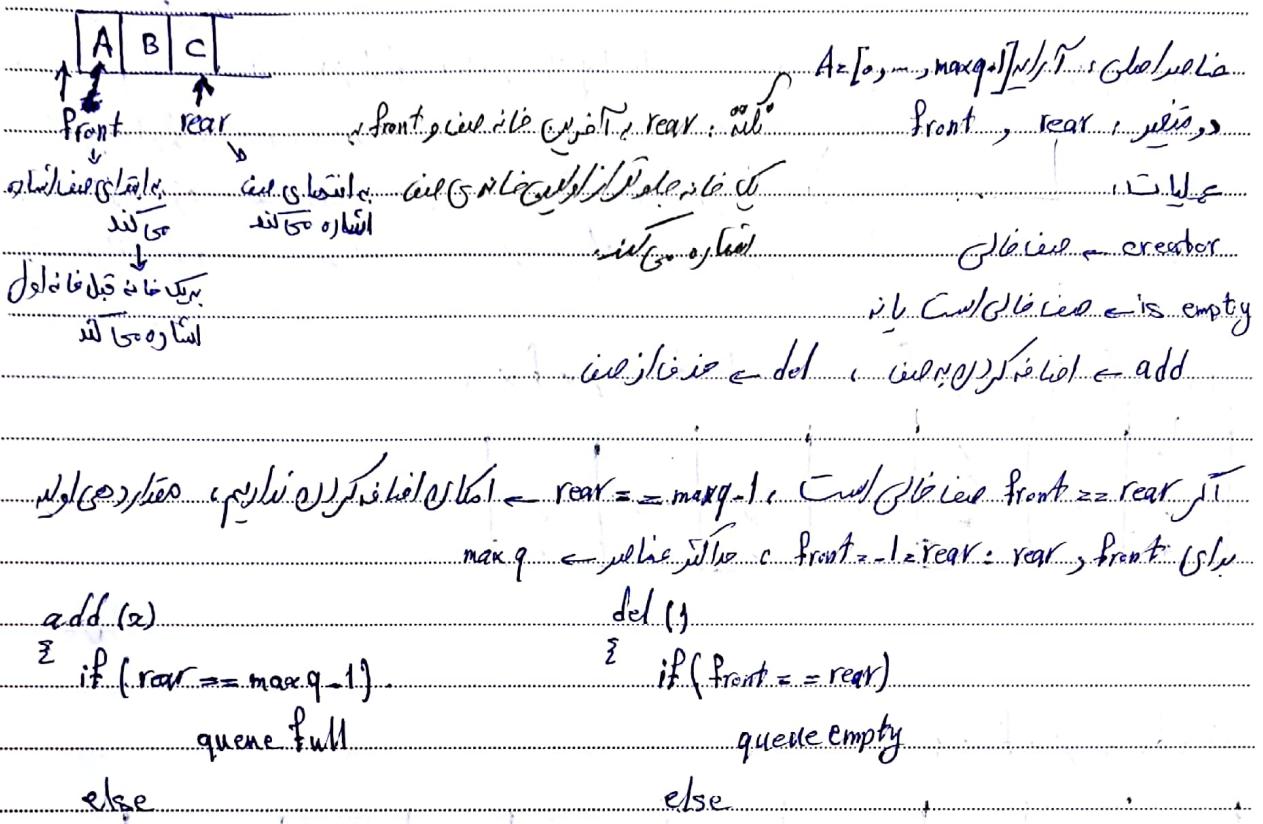
لای دیگر از استخراجی (رسی) باشند اینها همانند ساختار اطلاعاتی (data structure) هستند و معمولاً برای ذخیره داده ها (data) استفاده می شوند. اینها معمولاً برای ذخیره داده ها (data) استفاده می شوند.

Subject :

Date .....

FIFO queue & circ

مقدمة في الـ FIFO (First In First Out) و الـ Queue (Circular Queue)



فرانت (front) هو العنصر الأول في المقدمة (First).  
الخلف (rear) هو العنصر الأخير في المقدمة (Last).

maxq-1 ← العنصر الثاني ... front = rear ← العنصر الأول.  
 $(\text{rear}+1) \bmod (\text{maxq}) = \text{front}$  ← العنصر الثاني ... front = rear ← العنصر الأول.

$$\begin{aligned} F-R &= \text{العنصر الثاني} : F > R \\ R-F &= \text{العنصر الأول} : R > F \end{aligned} \quad \left. \right\}$$

Subject :

Date .....

add(x)

{ rear = (rear + 1) mod max\_q

if rear == front

queue full

else

A[rear] = x

5

del()

? if (front == rear)

queue empty

else

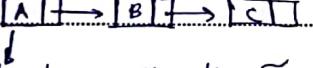
front = (front + 1) mod max\_q

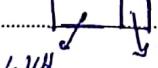
return A[front]

5

Qasr al-kutub (كتاب) و المكتبة العامة (الكتاب) في لندن إنجلترا  
جامعة بيركشاير (University of Berkshire)

مكتبة (كتاب) في لندن إنجلترا

  
front → A (أول)  
next → B (ثاني)  
next → C (ثالث)

  
null  
A (أول)  
B (ثاني)

struct node {  
 int info;  
 node\* next;

};  
node\* first, q  
new (first)  
first → info = A

first → next = null  
q = first  
for (i=1; i<=n; i++)

p = first  
first = first → next  
free (p)  
new (p)

p → info = List[i]  
p → next = null  
q → next = p

q = p

Subject :

Date .....

انشاء متغير لبيان اول  
براسن (رسن) اس

p = first  
while (p <> null)  
print (p->info)  
p = p->next

reverse LinkList ①

reverse a stack using another one ②

① reverse(node node, node pre, boolean flag):

pre = node;  
if (node->next != null)  
reverse (node->next, pre, flag)  
else  
flag = true  
node->next = pre

② if ( $n \times 2 == 0$ ) {

temp1 = stack1.top  
stack1.pop();  
for (j=1; j < stack2.size(); j++)  
stack1.push (stack2.top)  
stack2.pop();  
stack2.push (temp1)

(الخطوة 2)

عمل خط: node \*p

p = first  
while (p <> null)  
if p->info == value  
print ("—")  
break  
p = p->next.

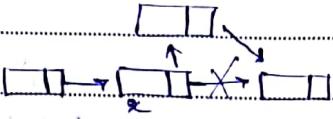
Subject :

Date .....

new(p)

p.info = value

p.next = x.next



x.next = p

if node == p: p = first first = first.next free(p)

elif node == first: p = first

while (p.next != x):

p = p.next

p.next = x.next

free(x)

pop(): removes previous element from stack & returns it as output

push(): adds element at top of stack

push(var):

new(p)

p.pop() if top == null

p.info = var

if top == null

p.next = top

p = top

top = p

top = top.next

print(p.info)

free(p)

add(first, rear, var): adds new element at end of list

if front == null: first = rear = p

add(first, rear, var):

new(p)

else:

p.info = var

rear.next = p

if front == null:

rear = p

Subject :

Date .....

del(front, rear) {

if front == null

p = ~~front~~ front

front = p → front → next

if front == null

rear = null

print(p → info)

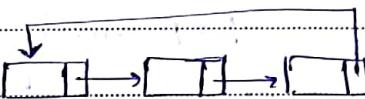
free(p)

ادخل العقد المطلوب

$x^2 + 2x + 4$



اذهب إلى العقد



ادخل العقد المطلوب

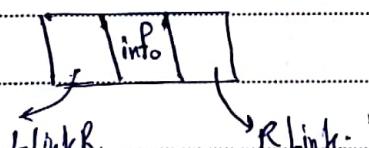
ومن أول العقد

new(x)

x → info = value

x → next = p → next

p → next = x



ادخل العقد المطلوب

new(p)

p → info = value

p → R.Link = x → R.Link

x → R.Link → L.Link = p

x → R.Link = p

x → R.Link → L.Link = x → L.Link

x → L.Link → R.Link = x → R.Link

ادخل العقد المطلوب

free(x)

MICRO

Subject :

Date .....

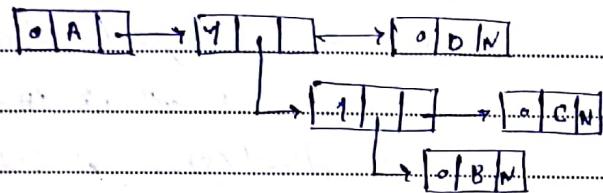


دليان (linked list)



نود (node)

linked list consists of node info



Baih (A, (B, C), D)

process(p){

if(p->null)

if(p->tag==0)

print(p->info)

else

process(p->info)

process(p->next)

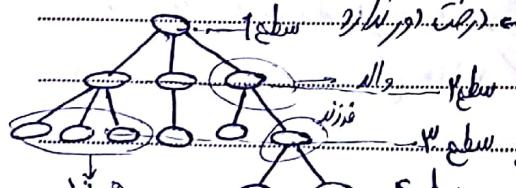
}

جذب (linked) tree

جذب غرافي

جذب لاسلكي (wireless) tree

جذب زيراني

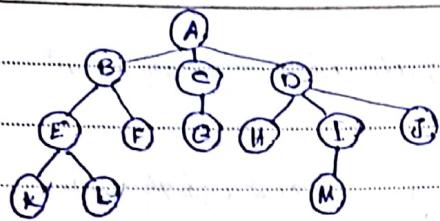


جذب بابا

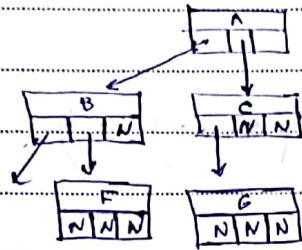
جذب ملائكة

Subject :

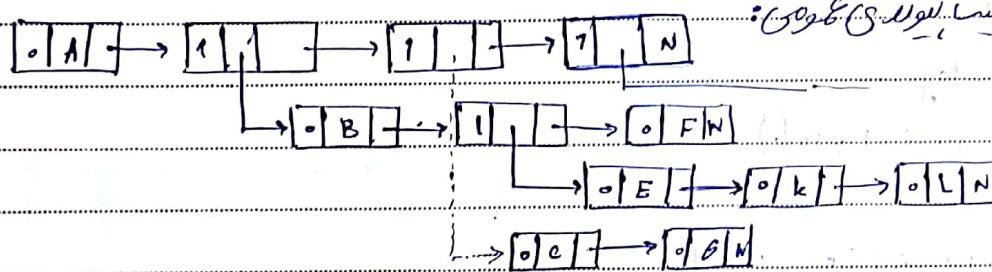
Date .....



(A(B(E(k,l),F),C(G),D(M(~~H~~M),I,J))

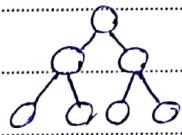


براده سازی بالاک لست  
در درخت بالا نود و درخت که خارج شود و درخت که خارج شود  
برای اینجا (C,N) است. (A,N-1) و (B,N) لستی است که خارج شود  
برای اینجا (C,N) است. (A,N-1) و (B,N) لستی است که خارج شود  
برای اینجا (C,N) است. (A,N-1) و (B,N) لستی است که خارج شود



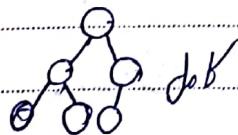
براده سازی بالاک لست

درخت دو دوی (binary) هی تواند تنهی بالاک عالو ببر رکته زیر درخت جب و در لست دار (عده ای که در فرزند دارد) در هر گز



درخت بی ری  
حالات که رکه ادارد

درخت کامل درخت کامل رکه (کام) در لست جب کام لست قرار گرفته بالاک  
و به عرض سطح کفر بقیه سطح های بالاک کام بالاک



در درخت دو دوی حالات که رکه ادارد (نودها) که موجود در لست کام =  $\frac{2^i - 1}{2}$

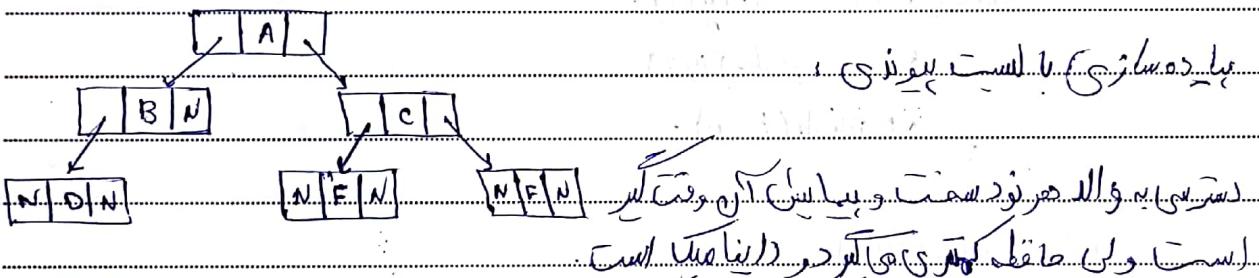
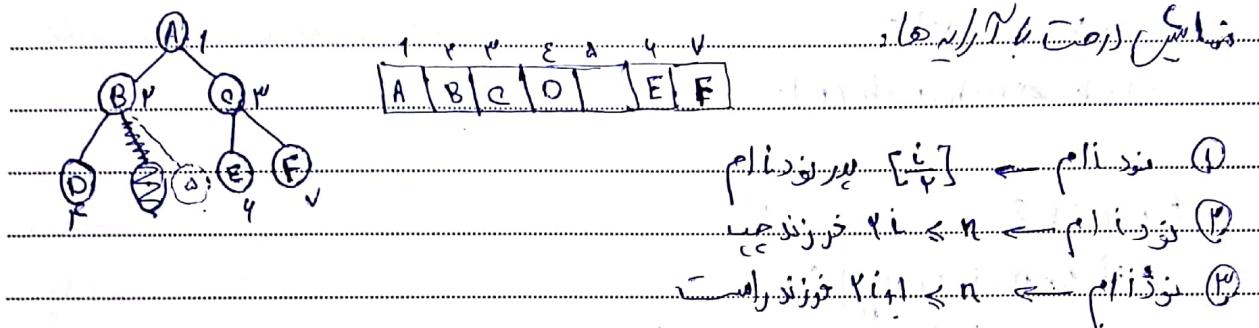
Subject :

Date .....

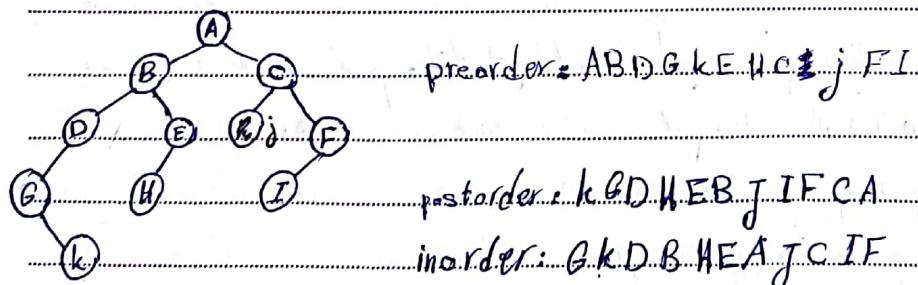
$$1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}$$

درخت باینری بروز رسانی با کمینه

$P^{k-1} =$  درخت باینری با عمق ک : تعداد نوادگان  $= 2^k$   
 $\lceil \log n \rceil + 1 =$  درخت باینری با عمق  $\log(n+1) =$  درخت باینری با عمق  $\lceil \log n \rceil$



LRV ← postorder LVR ← inorder VLR ← preorder



جستجو در درخت باینری با عمق ک

...stack.....push.c...pp;

void push(int v) {

stack.push(v);

```
int pop() {
```

if (pp.isEmpty) {

while( ! pp.isempty() ) {

PP push(t-temp);

```
return p.p.pop();
```

truth is also a lie, because it is not true. (وَالْحَقُّ كَاذِبٌ لَا يَرْجُعُ عَنْهُ إِنَّمَا يَرْجُعُ عَنِ الْكَاذِبِ)

مصحح کردار از پسوندیت لیگنیکوں کے لئے

الآن يسمى بـ (O<sub>(n)</sub>) ، وهذا يعني أن المقدار الذي يزيد عن المقدار المطلوب

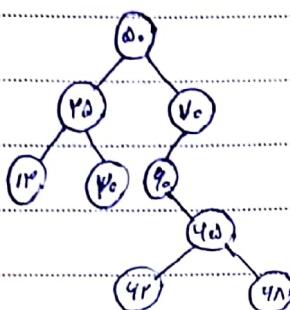
لطفاً <sup>(1)</sup> stack خالی می‌شود و از آنها آغاز می‌کرد. آنها می‌توانند اگر خالی بود آنها پیش روی اینها قرار گیرند. اگر خالی نبود با بالاترین stack علاوه بر آنها می‌توانند فرآوری را انجام دهند. اگر خالی نباشد stack آنها را باز کرده و آنها را در اینجا می‌گذارند. درینجا  $\text{stack}_1$  که باقی می‌ماند stack  $\text{stack}_2$  را در آن می‌گذارد.



1. (binary search tree) BST

null, if there is no left child, then it is a leaf node. If there is a right child, then it is a non-leaf node. (1)

(2) Leaf node (non-leaf node)



$O(n)$  جستجوی عصب و جویی  $O(n)$  جستجوی عصب و جویی  $O(n)$  جستجوی عصب و جویی

$O(n)$  جستجوی عصب و جویی  $O(n)$  جستجوی عصب و جویی  $O(n)$  جستجوی عصب و جویی

جستجوی لایست  $\leftarrow O(1)$  جستجوی عصب و جویی در درخت BST

BST-search(p, key)

if ( $p == \text{null}$ )

return ("not found")

if ( $p \rightarrow \text{info} == \text{key}$ )

return (p);

if ( $p \rightarrow \text{info} > \text{key}$ )

return (BST-search( $p \rightarrow \text{L Link}$ , key))

return (BST-search( $p \rightarrow \text{R Link}$ , key))

5

BST-insert(p, key)

q = null

p = root

while ( $p \neq \text{null}$ )

if ( $p \rightarrow \text{info} == \text{key}$ )

if ( $p \rightarrow \text{info} > \text{key}$ )

$p = p \rightarrow \text{L Link}$

else

$p = p \rightarrow \text{R Link}$

MICRO

return (false)

no overflow

Subject :

Date .....

new(x)

جذر اهم اعداد

x.info = key

باقی قرار دهم

x.R.Link = null



x.L.Link = null



if (root == null)



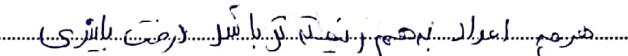
root = x



else



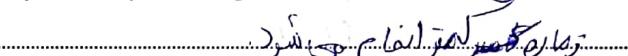
if (q.info > key)



q.L.Link = x



else



q.R.Link = x



return True

5

مسنون

برای جذب مسندی با هم فریاد کنیم

و حقیقی عضویت در فرزندان جذب می کنیم

BST.Del (root, x)

برای حذف یک عضو از درخت باید آن را پیدا کرد

if (x.R.Link == null  
or x.L.Link == null)

را جایگزین کنیم

q = null

وارد ساختی را بخواهیم که من خواهیم چندن اینم که شوام و میخواهد

p = root

بهم تغییر میکند

while (p <> x)

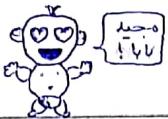
q = p

if (p.info < x.info)

p = p.R.Link

else

Subject :



Date .....

else {

    q = x

    p = x → R.link

    while (p → L.link ≠ null)

        q = p

        p = p → L.link

    if (p → R.link ≠ null)

        y = p → R.link

    else

        y = p → L.link

    if (q = null)

        root = y

    if (q ≠ R.link & q ≠ L.link)

        else {

            if p → info > q → info

                q → R.link = y

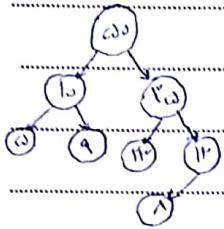
            else

                q → L.link = y

            if p != x

                x → info = p → info

}



: الیہ ایک جیسی لیہ جو جگہ سے بھی مکمل لیہ : Max Tree

اندھی لیہ جو جگہ سے بھی نہ مکمل لیہ : Min Tree

یہی جس پر ایک مکمل لیہ

اندھی لیہ جو جگہ سے بھی مکمل لیہ : Maxheap

(جیسے h = logn) اندازہ کاونٹ کرنے کے لیے جو جگہ سے بھی مکمل لیہ : Min Tree

O(bgn) : جیسے جو جگہ سے بھی مکمل لیہ

اندازہ کرنے کے لیے جو جگہ سے بھی مکمل لیہ : Maxheap

Maxheap insertion (H.l.n & key):

l = n+1

j = i  
while ( j > 0 and  $H[i] < H[j]$ )  
 $H[i] = H[j]$

i = j

j =  $\frac{i}{2}$   
 $H[i] = key$

ایسا جیسے جو فرستے براہی جسیں جائیں جائیں جو ادا کر دیں اور جو ادا کر دیں جائیں جو ادا کر دیں

Subject :

Date .....

Max heap - O(n)

key = H[1]

k = H[n]

i = 1, j = 1

→ 2

while (j < n)

if ( $H[j] < H[j+1]$ )

j++

if ( $k > H[j]$ )

break

$H[i] = H[j]$

i = j

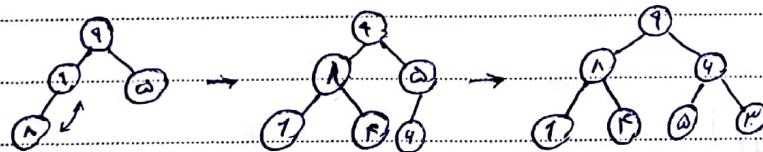
5 j = i + 1

$H[i] = k$

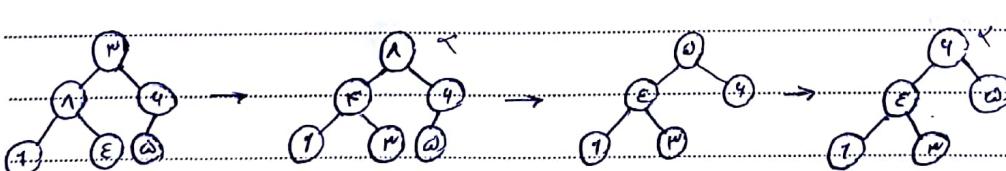
return(key)

9 1 0 A E 4 W

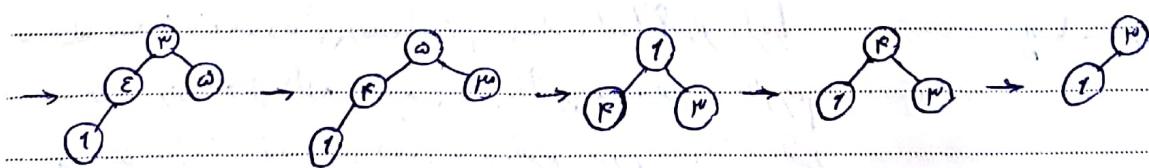
max heap (مکرر) و میکرر (معنی دارد)



21 ①



22 ①



23 ①

Subject :

Date .....

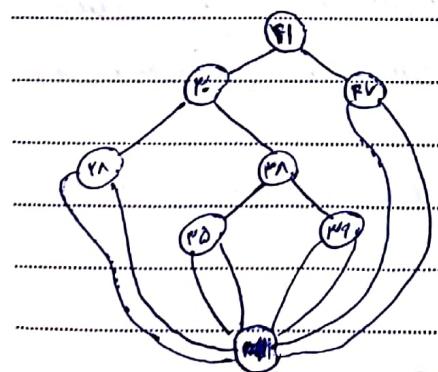
for  $i = 1$  to  $n$ max-heap-insertion ( $A[i]$ )  $\rightarrow O(n \log n)$  $\} \rightarrow O(n \log n)$ for  $i = 1$  to  $n$  $A[i] = \text{max-heap-del}(1) \rightarrow O(n \log n)$ النحوتة  $O(n \log n) \rightarrow \text{heap sort}$  ②النحوتة  $O(n \log n) \rightarrow \text{AVL tree} + \text{Inorder traversal}$  ③

DS	$O$	حذف	$O(N)$ حذف و $O(\log n)$ إنشاء heap
أريانا	$O(1)$	$O(n)$	Insert $O(n)$ حذف $O(n)$ إنشاء heap
أريانا	$O(1)$	$O(n)$	Insert $O(n)$ حذف $O(n)$ إنشاء heap
أريانا	$O(n)$	$O(1)$	Insert $O(n)$ حذف $O(1)$ إنشاء heap
heap	$O(\log n)$	$O(\log n)$	Insert $O(\log n)$ حذف $O(\log n)$ إنشاء heap

النحوتة  $O(n \log n)$  دليل الميزى دليل متوازن بورنج تفاصيل الميزى

B.S.T. [Link | inf | RLink]

R.B.T. [Link | Color | info | Parent | RLink]

النحوتة  $O(n \log n)$  دليل الميزى دليل متوازن بورنج تفاصيل الميزىblack  $\leftarrow$  null,  $bNull$  ④ black  $\leftarrow$  root ①newNode  $\leftarrow$  insert( $bNull$ ,  $newNode$ ) ⑤newNode  $\leftarrow$  fixHeight( $bNull$ ) ⑥newNode  $\leftarrow$  fixColor( $bNull$ ) ⑦ $bNull \leftarrow \text{newNode}$  ⑧ black height

MICRO

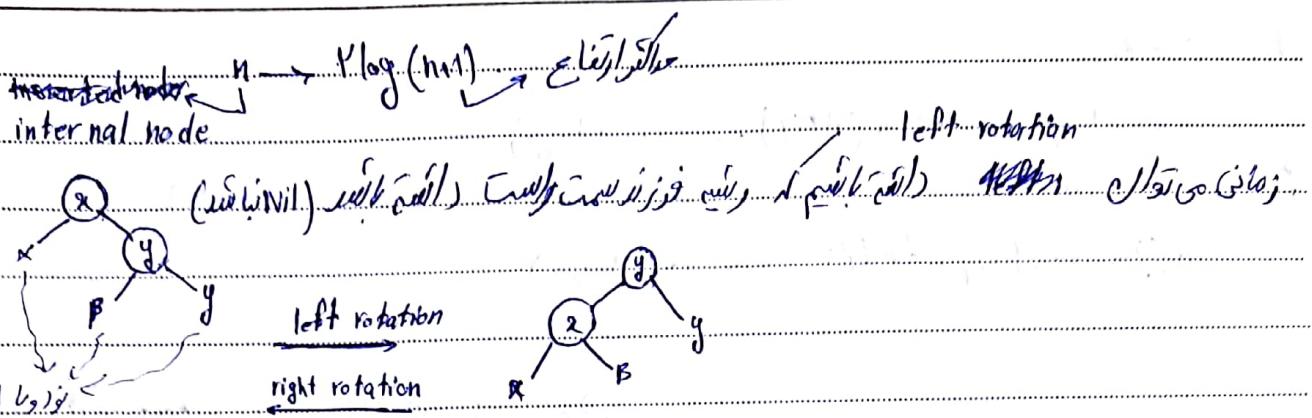
Null

End BH = Y

BH  $\leftarrow$  newHeight( $bNull$ )

Subject :

Date .....



left-rotation ( $T, x$ ):

$y \leftarrow \text{right}[x]$

$\text{right}[x] \leftarrow \text{left}[y]$

if  $\text{left}[y] \neq \text{nil}[T]$

then  $p[\text{left}[y]] \leftarrow x$

$p[y] \leftarrow p[x]$

if  $p[x] = \text{nil}[T]$

then  $\text{root}[T] \leftarrow y$

else if ( $x = \text{left}[p[y]]$ )

then  $\text{left}[p[x]] \leftarrow y$

else  $\text{right}[p[x]] \leftarrow y$

$\text{left}[y] \leftarrow x$

$p[x] \leftarrow y$

BST-insertion ( $T, z$ )

$\text{color}[z] \leftarrow \text{red}$

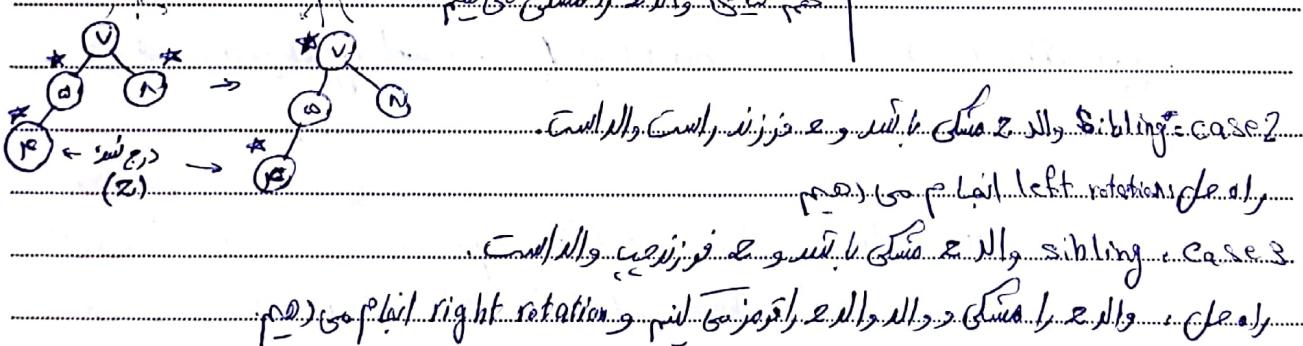
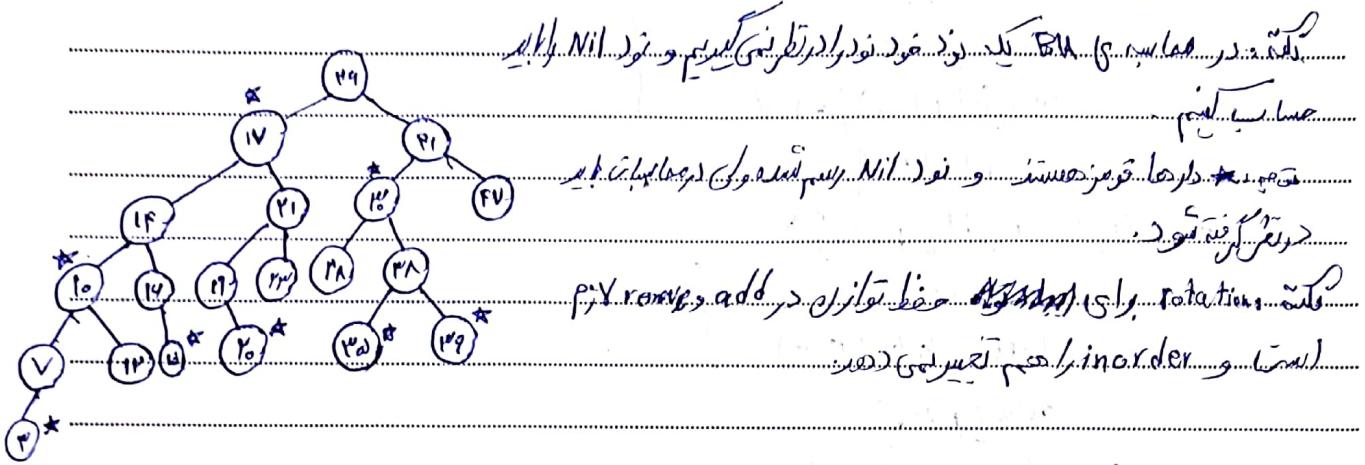
$p[z] \leftarrow \text{parent}$

R.B.-insertion-fixup ( $T, z$ )

**Subject :**

Date .....

R.B. insection lamp (r, z)



R.B.-insert&fixup(T,r,z)

.....while (color(p[2]) == red)

if ( $p[z] = \text{left}[p[p[z]]]$ )

$y = \text{right}[\rho[\rho[\text{EZ}]]]$

**Subject :**

Date .....

if (color[y] == red)

Color [p[ex]] = black

Color: #47z black

`Color[p[p[z]]] = red`

else  $Z = P[P[Z]]$

else if ( $x == \text{right}[p[x]]$ )

$$E = P[Z]$$

left-natale ( $T, z$ )

~~color of page 39~~

Color  $\text{P}_p[2]$  = black

color [ p [ p [ z ] ] ] = red

right-rotation( $\tau$ ,  $p[p[z]]$ )

else

(same as the clause with right and left exchange.)

color[root[ $\tau$ ]] = black

الورقة العاشرة (الطباعة) و سعر (rotation) =  $\log(n)$  و سعر (الطباعة) =  $\log(n)$

(١) درخت که نوع آن (الف) فوج درار (الف) فوج خنفری عورمهزی بالا است: قلادوئی  
نیفون (نیشن) (ب) فوج خنفری سینه: قلادوئی black boot و تولای درون (عورمهزی) کپسا (کپسا)  
تعداد: ۱۰۰۰ درجه درجه سکوت از این نیفون

کوئی خدا را حنف نمایند و مسیح را در آن خود که تصور خش را داشتند بخواهند

**■ Subject:**

Date:

<p>خزندنون (حصن) توزیع این <math>\leftarrow</math> <math>\rightarrow</math> <math>\leftarrow</math> <math>\rightarrow</math> <math>\leftarrow</math> <math>\rightarrow</math></p> <p>بعد از عرض خواهد توزیع شد</p> <p><math>\left\{ \begin{array}{l} \text{Case 1:} \\ \text{هنری در میان دو فرزند} \end{array} \right.</math></p> <p><math>\left\{ \begin{array}{l} \text{Case 2:} \\ \text{هنری در میان دو فرزند} \end{array} \right.</math></p> <p><math>\left\{ \begin{array}{l} \text{Case 3:} \\ \text{هنری در میان دو فرزند} \end{array} \right.</math></p> <p><math>\left\{ \begin{array}{l} \text{Case 4:} \\ \text{هنری در میان دو فرزند} \end{array} \right.</math></p>	<p>خزندنون (حصن) توزیع شد</p>
<p>هنری در میان دو فرزند</p>	<p>هنری در میان دو فرزند</p>

R B Dellet - fix up ( $T, x$ )

```

while ( $x \neq \text{root}[T]$  and  $\text{color}[x] = \text{black}$ )
    if ( $x == \text{left}[p[x]]$ )
         $w = \text{right}[p[x]]$ 
        if ( $\text{color}[w] == \text{red}$ )
             $\text{color}[w] \leftarrow \text{black}$ 
             $\text{color}[p[x]] \leftarrow \text{red}$ 
            leftrotation( $T, p[x]$ )
    }
}

```

■ Subject:

■ Date:

$w \leftarrow \text{right}[P[x]]$

}

if  $\text{color}[\text{left}[w]] = \text{black}$  and  $\text{color}[\text{right}[w]] = \text{black}$

{

$\text{color}[w] \leftarrow \text{red}$

$x \leftarrow P[x]$

}

else if  $\text{color}[\text{right}[w]] \neq \text{black}$

{ if  $\text{color}[\text{right}[w]] = \text{black}$  {

$\text{color}[w] \leftarrow \text{red}$

$\text{color}[\text{left}[w]] \leftarrow \text{black}$

right rotation( $T, P[x]$ )

$w \leftarrow \text{right}$

} case 3

$\text{color}[w] < \text{color}[P[x]]$

else case

$\text{color}[P[x]] \leftarrow \text{black}$

$\text{color}[\text{right}[w]] \leftarrow \text{black}$

left rotation( $T, P[x]$ )

$x \leftarrow \text{root}$

}

else (same as then clause with right and left exchanged)

*A.R.Mi.P.P.*

$\text{color}[x] \leftarrow \text{black}$

■ Subject:

■ Date:

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

order statistic

new structure with  $i$ th order statistic

each node has size field like size of RBT

key value changes each logn, size of tree is constant

L	key	color	parent	R	size
---	-----	-------	--------	---	------

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

for  $i < m$   $\rightarrow$  size[i]  $\leq i$   $\rightarrow$  size[i]  $\leq \frac{m}{2}$   $\rightarrow$  left[i]  $\leq \frac{m}{2}$   $\rightarrow$  right[i]  $\geq \frac{m}{2}$   $\rightarrow$  right[i]  $\geq i$   $\rightarrow$  right[i]  $\geq \frac{m}{2}$   $\rightarrow$  right[i]  $\geq i$

os-select( $x, i$ )

$$r \leftarrow \text{size}[\text{left}[x]] + 1$$

if ( $r = i$ )

return  $x$

else if ( $i < r$ )

return os-select(left[x], i)

else

return os-select(right[x], i - r)

Subject:

Date:

الآن  $\text{rank}(x)$  يساوي العدد البراري من العناصر التي تقع قبل  $x$  في الترتيب المترافق.

نريد أن نحسب  $\text{rank}(x)$  في  $T$ . إذا كان  $x$  هو root، فـ  $\text{rank}(x) = \text{size}(\text{left}[x]) + 1$ .  
إذا كان  $x$  ليس root، فـ  $\text{rank}(x) = \text{rank}(\text{right}[p[y]]) + \text{size}(\text{left}[p[y]]) + 1$ .

as  $\text{rank}(T, x) \{$

$r \leftarrow \text{rank}[\text{left}[x]] + 1$

$y \leftarrow x$

while ( $y \neq \text{root}[T]$ )

if ( $y = \text{right}[p[y]]$ )

$r \leftarrow r + \text{size}[\text{left}[p[y]]] + 1$

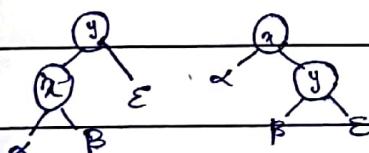
$y \leftarrow p[y]$

return  $r$

لذلك  $\text{rank}(x)$  يساوي العدد البراري من العناصر التي تقع قبل  $x$  في الترتيب المترافق.

نريد أن نحسب  $\text{rank}(x)$  في  $T$ . إذا كان  $x$  هو root، فـ  $\text{rank}(x) = \text{size}(\text{left}[x]) + 1$ .  
إذا كان  $x$  ليس root، فـ  $\text{rank}(x) = \text{rank}(\text{right}[p[y]]) + \text{size}(\text{left}[p[y]]) + 1$ .

نريد أن نحسب  $\text{rank}(x)$  في  $T$ . إذا كان  $x$  هو root، فـ  $\text{rank}(x) = \text{size}(\text{left}[x]) + 1$ .  
إذا كان  $x$  ليس root، فـ  $\text{rank}(x) = \text{rank}(\text{right}[p[y]]) + \text{size}(\text{left}[p[y]]) + 1$ .



$$\text{size}[y] \leftarrow \text{size}[x] \quad \text{size}[x] \leftarrow \text{size}[\text{left}[x]] + \text{size}[\text{right}[x]] + 1$$

الآن  $\text{rank}(x)$  يساوي العدد البراري من العناصر التي تقع قبل  $x$  في الترتيب المترافق.

■ Subject:

■ Date:

يسار و میان و راست درخت اوردر استریکس  $\Rightarrow$  ریشه

این دو دسته دیگر کلیه دیگر دو دسته هستند  $\Rightarrow$  این دو دسته هستند

interval tree

$$[t_1, t_r] \subset [t_1, t_p] \subset t_1 \leq a < t_p : [t_1, t_r] \subset t_1 \leq a < t_r : [t_1, t_r]$$

if ( $\text{low}[i] \leq \text{high}[i']$  and  $\text{low}[i'] \leq \text{high}[i]$ )

then ( $i \vee i' + \phi$ )

برای interval tree

①  $i \neq i' \neq \phi$  ②  $i = \text{left } i'$  ③  $i = \text{right } i'$

پس از max  $\rightarrow$  ایجاد interval tree  $\rightarrow$  ایجاد red-black tree  $\rightarrow$  interval tree

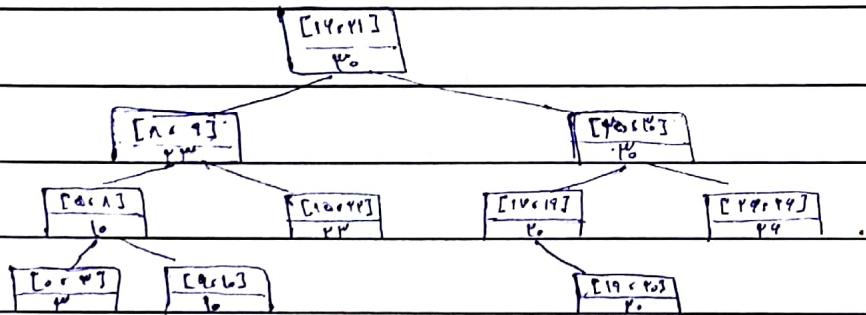
از هر دوی ایجاد

L	P	key	color	R
---	---	-----	-------	---

 $\rightarrow$  ایجاد،  
 $i \rightarrow [t_1, t_r], t_1 \rightarrow \text{low}[\text{int}[i]] < t_r \rightarrow \text{high}[\text{int}[i]]$

$\text{key}[x] = \text{low}[\text{int}[x]]$   $\rightarrow$  max  $\rightarrow$  max  $\rightarrow$  max

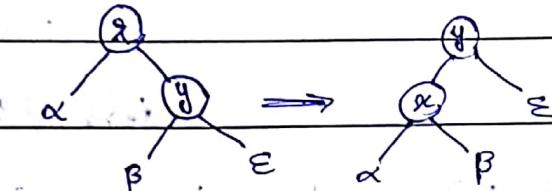
$\text{max}[x] \leftarrow \text{max}[\text{max}[\text{left}[x]], \text{max}[\text{right}[x]], \text{high}[\text{int}[x]]]$



## Subject:

Date:

② : (left) rotate  $\rightarrow$  fix-up case 2  
 ④ : (right) rotate  $\rightarrow$  fix-up case 4



: (obj) → rotate, fix-up (obj)

$$\max [y] \leftarrow \max [x]$$

$\max[x] \leftarrow \max(\max[\text{left}[x]], \max[\text{right}[x]])$  if  $\text{high}[x] < \text{low}[x]$

وَهُوَ مُنْسَبٌ إِلَيْهِ red black tree is

■ Subject:

Date:

iri' overlaid with another overlaid audio layer (so it has to play over the first one).

$$\text{low}[\text{int}[i]] < \text{high}[\text{int}[i']]$$

~~$\cdot \text{low}[\text{int}[i']] < \text{high}[\text{int}[i]]$~~

### interval-search(F[i])

$i \leftarrow \text{root}(\Gamma)$

`while( $x \neq \text{nil}$  [T], and tree does not overlap int [x])`

`if (left[x] ≠ nil[T] and low[i] < max[left[x]])`

$x \leftarrow \text{left}[x]$

else

~~$x \leftarrow \text{right}[x]$~~

return 2

۳

جافا

لما تم توليد المكونات كلها في  $G$ ، يُعرف المكون  $G$  بالرسوم  $G = (V, E)$

العنوان (E.E.C) يعني سعر وحدة طن متر مكعب (طن) وتحدد في العقود.

مُؤْكِدٌ (أ) بـ (ب) باز خود (نار) (ج) باز خود (نار)

دراست در فیزیک معمولی از آن است که  $n(n-1)$  درجه حریق داشته باشد.

- مترافق (Fully connected) : كل عقدة متصلة بجميع العقد الأخرى
- مترافق ضعيف (Weakly connected) : كل عقدة متصلة ببعض العقد الأخرى
- مترافق جزئي (Partially connected) : بعض العقد متصلة ببعض العقد الأخرى
- مترافق محدود (Limited connected) : بعض العقد متصلة ببعض العقد الأخرى
- مترافق متصل (Connected) : كل عقد متصلة ببعض العقد الأخرى
- مترافق غير متصل (Disconnected) : لا توجد روابط بين العقد

$$\text{length} = |E| \leftarrow \frac{1}{y} \sum_{i=1}^n d_i^y$$

دیگر: لایه لایه (لایه لایه) و بروک (بروک) و لایه (لایه)

برای خود سازی معرفت از (روش انتقالی) (مکانیزم) (۱) میتواند مجاورت

$\tau_{ij}$  يعطى عدد الأفراد في المجموعة  $i$  و  $j$  التي لها نفس الميزة  $m$  من المجموعات  $S_i$  و  $S_j$

ویرایش (چندین جمله) می‌تواند در اینجا مذکور شود.

$\sum_i a_{ui} = \deg(u)$  *يُمْكِنُ حِلُّ مُسَأَلَةِ الْمُدْرَسَةِ بِعِصْمَتِ الْمُكَوَّنِ*

$$\sum_i a_{iu} = \deg(u) \rightarrow \text{Count of edges} = |E|$$

جواب ایجاد کننده دارای  $a_{ij}$  است که مقدار  $V_j$  را در  $V_i$  افزایش داده باشد.

■ Subject:

■ Date:

دستگاه مکانیکی با این نسبت دارای اندازه  $\sum r_i = EI$  است و جمع اندامات آن را می‌توان با این نسبت محاسبه کرد.

برای محاسبه میانگین طول میانگین میانگین نویسندگان این روش را در پیشنهاد نموده اند.

آنچه در اینجا مذکور شده است

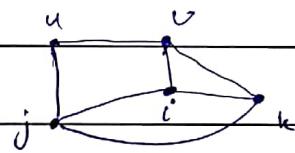
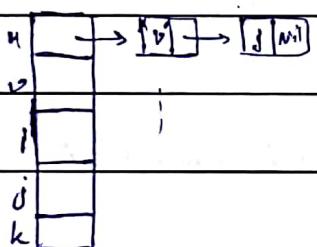
آنکه این نسبت میانگین طول میانگین میانگین نویسندگان را در پیشنهاد نموده اند.

این نسبت

آنکه این نسبت میانگین طول میانگین میانگین نویسندگان را در پیشنهاد نموده اند.

آنکه این نسبت میانگین طول میانگین میانگین نویسندگان را در پیشنهاد نموده اند.

آنکه این نسبت میانگین طول میانگین میانگین نویسندگان را در پیشنهاد نموده اند.



آنکه این نسبت میانگین طول میانگین میانگین نویسندگان را در پیشنهاد نموده اند.

آنکه این نسبت میانگین طول میانگین میانگین نویسندگان را در پیشنهاد نموده اند.

آنکه این نسبت میانگین طول میانگین میانگین نویسندگان را در پیشنهاد نموده اند.

آنکه این نسبت میانگین طول میانگین میانگین نویسندگان را در پیشنهاد نموده اند.

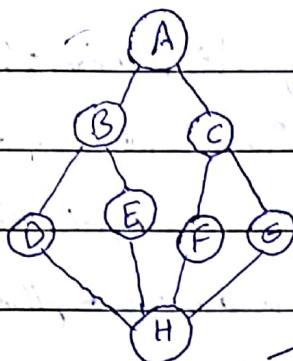
آنکه این نسبت میانگین طول میانگین میانگین نویسندگان را در پیشنهاد نموده اند.

Preorder (DFS) depth first search

BFS (BFS) breath first search

■ Subject:

Date:



١: سفر \* زمان شروع للحوار است (بخارى روى)

DFS(A): A, B, D, H, E, F, C, G

Cardio

For  $i = 1$  to  $n$

Time:  $O(n)$

DFS(G, v) {

visited[V] = 1;

for each  $u$  adjacent to  $V$  do

if  $\text{visited}[u] = 0$ ;

۲

A, B, C, D, E, F, G, H : ملکیت این چیز: قبضه: ①

ابد ایسخون (ایسخون) است (ایسخون) است (ایسخون)

آموزه حافظه ای برای مباحث ریاضیات  
لر رنگ ترین مراد ها از میری ترین حافظه های اندیشیدن

■ Subject:

■ Date:

breadth first search depth first search

Code:

for i=1 to n

Visited[i]=0;

BFS(G,V){

Visited[V]=1;

Add queue(V)

while(!isEmpty(q)){

u=del\_queue(q)

for each i adjacent to V

if Visited[i]=0;

Visited[i]=1;

Add queue(q,i)

ویلیمیجی ایجاد کردن پیکل و ویلیج

3 num=0

for i=1 to n

if Visited[i]==false

Dfs/Bfs(G,i)

num++;

دسته بندی این مطلب

شناسنامه DFS, BFS

و درخت گراف

لی

| Subject:

Date:

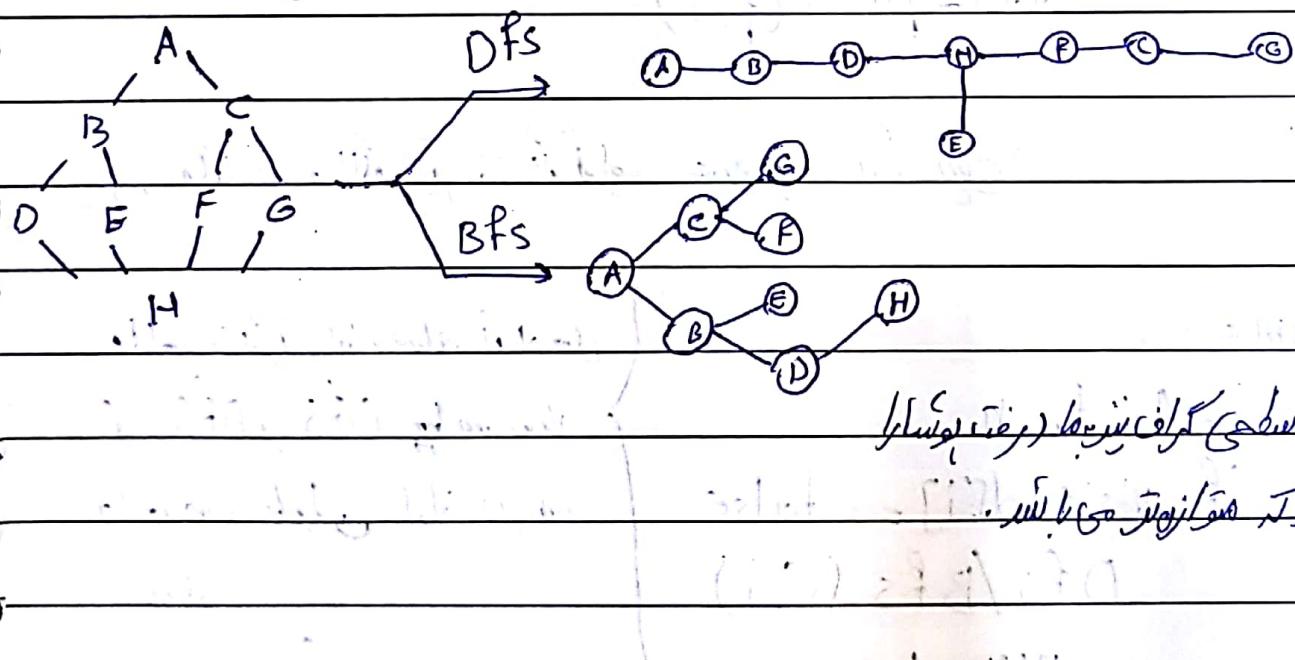
الشجرة الممتدة (Spanning Tree) هي شجرة ممتدة تربط جميع المنشآت في الشبكة.

نودهای کاریکاتوری جزو تعریفی اینهاست.

لصق استكرايل (EVA) و فرطان (PVC) و خواص متميزة

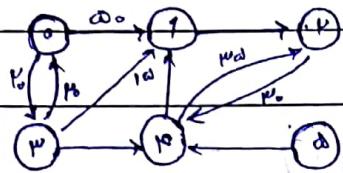
$E' \subset E$ ,  $V' = V$ ,  $|E'| = n-1$  (zugewiesen) !) ein zugesetztes

\* النحو والصرف (بايبر) در نظر عینتی سید (معذرة) طاهر حسین



■ Subject:

■ Date:



مُنْهَج دِيْكْسْتَرْ (Dijkstra's algorithm)  
مُنْهَج دِيْكْسْتَرْ يُعَدُّ مُنْهَجَيْنِ دِيْكْسْتَرْ وَبَرْكْهُولْدْ (Bellman-Ford)  
وَهُوَ مُنْهَجٌ يُعَدُّ مُنْهَجَيْنِ دِيْكْسْتَرْ وَبَرْكْهُولْدْ

(Dijkstra's algorithm is a greedy algorithm that finds the shortest path between nodes in a graph with non-negative edge weights.)

حيث يُعرَفُ بـ "الجهة" (edge) بـ "الوجه" (face) في المثلثات  
في هذه الحالة يتم إيجاد المسافة من الأصل (source) إلى كل (node)

طريق من الأصل (source) إلى كل (node)

Dijkstra-shortestPath( $n, v$ )

for ( $i = 0, i < n, i++$ ) {

$s[i] = \text{false}$ ;

$dist[i] = \text{length}[v][i]$

}

        ↓  
        ما هي المسافر

$s[v] = \text{True}$

$dist[v] = 0$

    for ( $i = 0, i < n, i++$ ) {

$u = \text{choose}(n)$

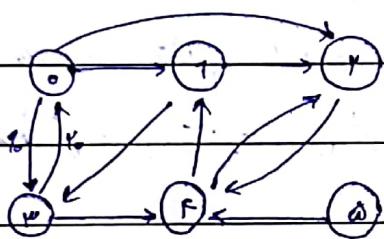
$s[u] = \text{True}$

( $S$  هو

■ Subject:

■ Date:

```
for(w=0, w<n, w++) {  
    if(!S[w]) {  
        if(dist[u] + length[u][w] < dist[w])  
            dist[w] = dist[u] + length[v][w].  
    }  
}
```



0	1	2	3	4	5
T	F	F	F	F	F

dist =	0	00	00	10	M	M
S =	T	F	F	T	F	F

dist =	0	00	00	10	10	M
S =	T	F	F	T	T	F

dist =	0	00	00	10	10	10	M
S =	T	F	F	T	T	F	M

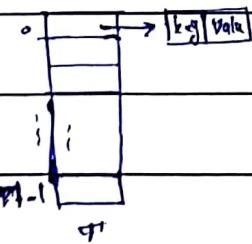
$O(n^2)$  time complexity

with this we can search the key value in Hash table  
جامعة كل المفاتيح

Hash table = direct accessible

in Hash table key :  $k_1, k_2, \dots, k_n$ ,  $|V| = |U|$ , so value of  $v_i$  will be  $v_i$  and key  $k_i$  is also  $v_i$

universal keys



key value of  $v_i$  is called index of  $v_i$ .

$x \rightarrow [k_0 | 'A']$

ARMinSP

■ Subject:

■ Date:

insert ( $T, x$ )

$$T(\text{key}[x]) \leftarrow x$$

delete ( $T, x$ )

$$T(\text{key}[x]) \leftarrow \text{null}$$

search ( $T, k$ )

return  $T[k]$

وقت اقصى  $O(1)$  و  $O(n)$  في الحالات

hash table if  $\text{key} \in \text{list}$ ,  $\text{add} \rightarrow \text{hash function} \rightarrow \text{Hash Table}$

if  $\text{key} \notin \text{list}$   $\rightarrow$  insert  $\text{key}$   $\rightarrow$   $\text{create new list} \rightarrow \text{append to Hash Table}$   
open addressing (1) chaining (2)  $\rightarrow$   $\text{best case} = O(1)$   $\text{worst case} = O(n)$

Chaining  $\rightarrow$   $\text{key} \in \text{list}$ ,  $\text{list} \rightarrow \text{index} \rightarrow \text{key}$   $\rightarrow$  chaining  
 $\rightarrow$   $\text{best case} = O(1)$   $\text{worst case} = O(n)$

insert ( $T, m$ )

insert  $m$  at the head of the list  $T[h(\text{key}[m])] \rightarrow O(1)$

delete ( $T, x$ )

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

Worst Case:  $O(n)$

search ( $T, x$ )

search an element with key in list  $T[h(k)]$ .  $x \leftarrow \text{list}[h(k)]$   
 $\downarrow$   $O(n)$

$n/m$   $\rightarrow$   $\text{average case} = O(1)$   $\rightarrow$   $\text{worst case} = O(n)$   $\rightarrow$  hash function  $\rightarrow$   $\text{load factor} = n/m$

only delete,  $\rightarrow$   $\text{load factor} = 1$   $\rightarrow$   $\text{worst case} = O(n)$   $\rightarrow$   $\text{load factor} = n/m$   
 $\rightarrow$   $\text{worst case} = O(n)$   $\rightarrow$   $\text{load factor} = n/m$

Subject:

Date:

multiplication, Division : hash function

Division:  $h(k) = k \mod m$   $\rightarrow$   $k \in P^m$   $\Rightarrow$   $k \in \{0, 1, 2, \dots, m-1\}$

multiplication:  $h(k) = \lfloor m(kA \bmod 1) \rfloor$

$0 < A < 1$ ,  $kA$   $\in [0, 1)$ ,  $\lfloor kA \rfloor \in \{0, 1, 2, \dots, m-1\}$

الى index  $i$   $\in [0, m-1]$  key  $\rightarrow$   $g[i]$   $\in$  hash function  $\rightarrow$  open addressing

الى  $i$   $\in [0, m-1]$   $\rightarrow$   $T[i] \leftarrow k$   $\rightarrow$   $T$   $\in$  hash table

key  $\rightarrow$   $i$   $\in [0, m-1]$   $\rightarrow$   $i$   $\in$  index  $\rightarrow$   $i$   $\in$   $T$

فرزه کردن (ورزیم) (sort)

insert ( $T, k$ ):  $\Sigma$

$i \leftarrow$

index  $\in [0, m-1]$ , key  $\rightarrow$   $g[i]$   $\in$  hash function  $\rightarrow$   $i$   $\in$   $T$

repeat

$j \leftarrow h(k, i)$

if ( $T[j] == \text{null}$ )

$T[j] \leftarrow k$

return  $j$

$i \leftarrow$

until  $i == m$

print ("hash table overflow")

$\Sigma$