

**Subject :**

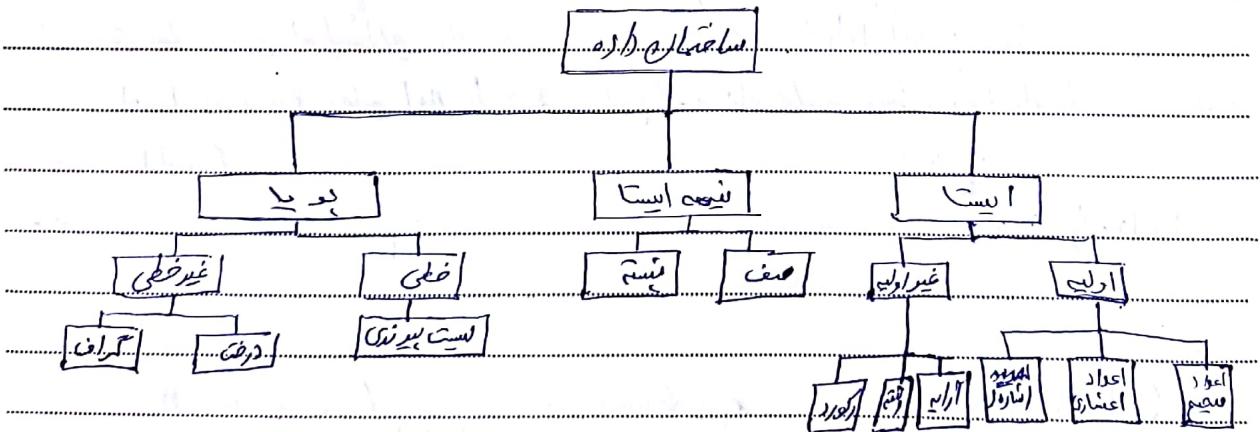
**Date** .....

CIRS < Harowitz 1 "لِجَانِيَّةِ الْمُؤْمِنِينَ" (جَنَاحُ الْمُؤْمِنِينَ) بـ "جَنَاحٌ مُّعَظِّمٌ"

ساخته از دو لایه

الله يرحمه: مجموعات از مجموعاتی که در آنها مذکور شده است  
درینه می تواند این را بخواهد که می خواهد و درینه می خواهد  
فرموده باشد که می خواهد و درینه می خواهد  
و درینه می خواهد و درینه می خواهد

وَالْمُؤْمِنُونَ إِذَا قَاتَلُوكُمْ إِذَا هُمْ مُّهَاجِرُونَ لَا يُعَذِّبُنَّكُمْ إِنَّ اللَّهَ يُحِبُّ الْمُتَّقِينَ



Subject :

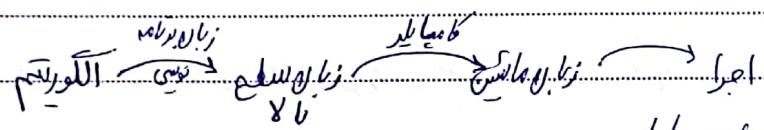
Date .....

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 گوریتم تمام گذشتند

برای بحث درست مبارزه با لایم (جیمز بورن) و مجموع این اجزای این نوع زمان برای این امور

و نوع ساده و مرتبت این روش کار (جیمز بورن)

function:  $f(1) \rightarrow 1$

int a  $\rightarrow 1$

$a = 0$

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

$a++$

return a

نحوی اینجا مدارک و موارد زمانی اینجا اینجا اینجا

۱) اینجا مدارک و موارد زمانی اینجا اینجا اینجا

۲) اینجا مدارک و موارد زمانی اینجا اینجا اینجا

return

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

نحوی اینجا مدارک و موارد زمانی

نحوی اینجا مدارک و موارد زمانی

$x = 0$

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

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

**Subject :**

Date .....

$$x_{2,0} = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n 1 \cdot z^{N^{10}}$$

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

for (j=1; j < n; j++)      untuk i,j,l,p,q       $\sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n 1 = n \left( \frac{n(n+1)}{4} \right)$

for ( $k=1$ ;  $k \leq n$ ;  $k++$ )

$$8+9 \quad \text{لـ} \sum_{i=1}^n \sum_{j=1}^l \sum_{k=1}^d 1 = \sum_{i=1}^n \sum_{j=1}^l j$$

$$= \sum_{i=1}^n \frac{i(i+1)}{r} = \frac{1}{r} \left[ \sum_{i=1}^n i^r + \sum_{i=1}^n i \right] = \frac{1}{r} \left( \frac{n(n+1)(rn+1)}{r} \right) \left( \frac{n(n+1)}{r} \right)$$

$$x=1, \quad l=1 \quad \frac{n}{1} \quad \frac{1}{1} \quad x=1$$

$\text{while } (i < n) \{$        $i$        $i$        $i^{x+1} < n \rightarrow k+1 < \log n \rightarrow k < \log n + 1$

$\sum_{i=1}^{244} f_i$   $\rightarrow$   $\bar{f}$   $\rightarrow$   $\bar{f}_i$   $\rightarrow$   $\bar{f}_{i+1}$

return  $n \cdot k^{\frac{1}{k-1}} < n$

2-1 i-1

$$x_{k+1} \leq \frac{n}{k+1} \leq k \log n + 1$$

$$S = \frac{i}{\gamma} \cdot \frac{1}{\mu} \cdot \frac{n}{k} \cdot \gamma^{n-1}$$

return n

for (int i=1; i<N; i++)

$$j=1 \quad \quad \quad 1 \quad \quad \quad 1 \quad \quad \quad n$$

$$\text{while } (j \leq i) \{ \quad \quad \quad f_j \quad p_j : \quad \sum_{i=1}^n \lceil \log_{f_r} i + 1 \rceil$$

X4.4

$$J = J_\nu \alpha^\nu \quad n = \frac{\log N_A}{\delta_\nu}$$

Subject :

Date .....

$x =$	$i \rightarrow \frac{n}{r} \leftarrow n$	$i$	$x$
$\text{for } (i=1; i < n; i++)$	$i$	$\frac{n}{r}$	
$\quad \text{for } (j=1; j < n; j++)$	$j \rightarrow \frac{n}{r} \leftarrow \frac{n}{r}$	$j$	$\frac{n}{r}$
$\quad \quad n--$	$j \rightarrow \frac{n}{r} \leftarrow \frac{n}{r}$	$j$	$\frac{n}{r}$
$\quad \quad x++$	$j$	$j$	$\frac{n}{r}$
$\text{return } x$	$k$	$\frac{n}{r^k}$	
$\text{Cm: } n \left( \frac{1}{r} + \frac{1}{r^2} + \frac{1}{r^3} + \dots \right) = n$			

$x =$	$i$	$x$
$\text{for } (i=1; i < n; i++)$	$1$	$n$
$\quad \text{for } (j=1; j < n; j++)$	$r$	$n-1$
$\quad \quad x++$	$r$	$n-r$
$\quad \quad n--$	$\frac{n}{r}$	$n - (\frac{n}{r})$
$\text{return } x$	$= \frac{n^r}{r} - \sum_{i=0}^{n-1} i = \frac{n^r}{r} - \frac{(n-1)n}{r}$	

insertion sort ( $A, n$ ) $\text{for } (k:=2 \text{ to } n) \rightarrow n$  $\text{key} = A[k] \rightarrow n-1$  $i = k-1 \rightarrow n-1$  $\text{while } (i > 0 \text{ and } A[i] > \text{key}) \rightarrow$  $A[i+1] = A[i]$  $i = i-1$  $A[i+1] = \text{key}$  $}$ 

$$f(n) = n + (n-1) + (n-2) + \sum_{k=r}^n t_k$$

$$+ r \sum_{k=r}^n (t_k - 1) + (n-1)$$

اجمالي التكاليف  $\leftarrow t_k$ while ( $i > 0$ )  $\leftarrow t_k$ 

مقدار التكاليف  $\leftarrow t_k = 1 : \text{الخطوة الأولى}$   
 مقدار التكاليف  $\leftarrow t_k = N : \text{الخطوة الأخيرة}$

$$\frac{1}{n!} \sum_{i=1}^n \sum_{k=r}^n t_k = \sum_{k=r}^n \sum_{i=1}^n \frac{t_k}{n!}$$

$\underbrace{\qquad\qquad\qquad}_{t_k}$

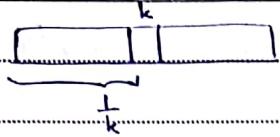
MICRO:  $\frac{1}{n!} \sum_{i=1}^n \sum_{k=r}^n t_k = \sum_{k=r}^n \frac{t_k}{n!}$

Cm:  $\frac{1}{n!} \sum_{i=1}^n \sum_{k=r}^n t_k = \sum_{k=r}^n \frac{t_k}{n!}$

الخطوة الأولى  $\leftarrow n!$

Subject :

Date .....

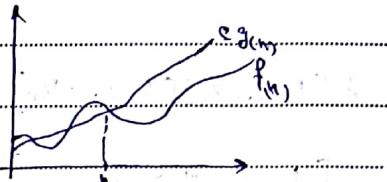


$$T_k = \sum_{j=1}^k \frac{1}{k} (k \cdot j + 1)$$

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

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

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  $\text{matrix}[U - L_r + 1] \times N$

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

$A[L_1 \dots L_r][U_1 \dots U_r]$  as type  $(U_r - L_r + 1) \times (U_r - L_r + 1)$

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

After shifting indices

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

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

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

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

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

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

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

جواب داده شد

$m \times n \times k$  کو  $C_m \times A_{m \times n} \times B_{n \times k}$  می نویسند

**Subject :**

Date .....

$(A_{\min} \alpha B_{\max}) \times C_{\max}$  (S.  $\min$   $\alpha$   $\max$   $\beta$   $\times$   $\max$   $\gamma$ )

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

$$\begin{array}{c} \text{Left side: } \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} R & R & R \\ 1 & F & M \\ R & R & M \\ R & 1 & M \end{bmatrix} \\ \text{Right side: } \text{Endfield (Syl.) JP lies.} \end{array}$$

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

LIFO  $\rightarrow$  Last In, First Out  
Block is empty. Block is created.  $\rightarrow$  LIFO top will stack layers.  $\rightarrow$  LIFO cyclic.

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.....menstack, g1, g2, ld, stack, else.....(6)

top = top + 1; if (top <= 1) item.push(); return stack[top];

5 stack [top] = item  
item = stack [top]

**top = top - 1**      return item

© 2010 Pearson Education, Inc., publishing as Pearson Addison Wesley.

Subject :

Date .....

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



$$\begin{array}{c} B \\ \sqsubseteq \\ C \\ \sqsubseteq \\ S \end{array}$$

۱) خارج از توابع

۲) ارزیابی عبارت

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

$AB + \leftarrow \text{postfix}_R$  روش سیوندی

$+ AB \leftarrow \text{prefix}_L$  روش پنوندی

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

$$axb+cx-ad \rightarrow ((axb)+c)-(ad)$$

ایجاد پر کردن

خود قرار گیرد

خود قرار گیرد

$$\xrightarrow{\text{پیشوند}} + ab.c / ad$$

۳) ای

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

امیر محمد نظری

الgoritم دارای دو مرحله است: مرحله انتخاب (selection) و مرحله ترتیب (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: مرحله انتخاب (selection) و مرحله ترتیب (sort).

مرحله انتخاب (selection) دارای  $n-1$  مرحله است.

از مرحله انتخاب (selection) نیز  $n-1$  مرحله است.

برای این مرحله انتخاب (selection) دارای  $n-1$  مرحله است.

و در مرحله انتخاب (selection) دارای  $n-1$  مرحله است.

۴) 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

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 .....

لکا (لئے) از (معا) میں لئیں گے۔ میر وہی کوں

$$\lg(\lg^{\alpha} n) \leq \lg^{\alpha} n \leq (\lg n)^{\alpha} = n^{\alpha/\ln 2}$$

$\lg(n!)$        $n^{\gamma n}$        $n^{\lg n}$        $\ln \ln n$        $\lg^* n$        $\mu \times n^{\gamma}$        $n^{\lg \lg n}$        $\ln n$       1

$$g^{lg n} \quad (\lg n)^{lg n} \quad e^n \quad n^{lg n} \quad (n+1)! \quad \sqrt{lg n} \quad \lg^*(lg n) \quad 4^{\sqrt{lg n}}$$

$$n \cdot \mu^n \cdot n! g_n \cdot \mu^{\mu^{n+1}}$$

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

$$(lg n)^r \quad (\lceil r \rceil)^{lg n} \quad \frac{n}{lg n} \quad \frac{n lg n}{lg n!} \quad \frac{lg n}{n^r} \quad n^w \quad (lg n)! \quad \frac{(lg n)^{lg n}}{lg lg n}$$

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

با شروع از درخته اول آن و در هر دفعه یک کوتاه باشد و با توجه به مقدار insertion rate  $\alpha$  می تواند قابلیت حذف کوچکتر از آن بوده اما در این قدری (هر دفعه یک نویم) باشد که این عیوب را برداشته باشند.

کو حلزونی صفت کر نماید اگر اراده استفاده از آن باشد.

[en]p) hub (bu:ylə) la:xis ang: inf:ix, pre:fix, post:fix, : üli:

مثلاً في الكلمة stack (قرارض) postfix (نهاية) intfix (في وسط).

**Subject :**

Date .....

فَلَمَّا نَهَى اللَّهُ عَنِ الْمُحَاجَةِ أَوْ لَوْجَاهَ بِالْأَثْرَى أَزْكَلَهُ (أَنْتَ)

.....میرا علی.....

## ۱- خروجی

نحوه مکمل، اینجا بخوبی لفظ است اگر او لوست بالاترین از top words (غیر اصلی) می شود.

كفر بالله ولهم عذاب أليم لأن حذفوا ومحى معهم أسم الله وله ملائكة

میر افغان ایز جاں (۱۴)

رَأْيُهُ مُؤْكِدٌ بِالْمُسَنَّدِ وَالْمُعْتَدَلِ وَالْمُسْتَقْبَلِ وَالْمُسْتَقْبَلِ وَالْمُسْتَقْبَلِ (١٢)

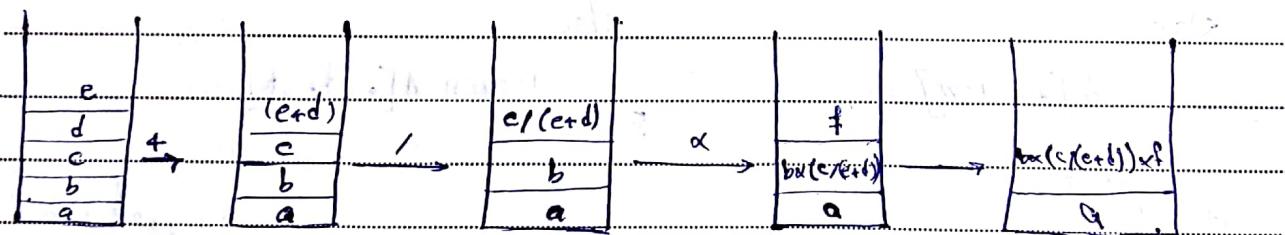
حَسْبَنِي اللَّهُو

$$a.b.c.(c/(d+e)).x.f \rightarrow abcde/f \times fx +$$



١٦

ab cde a f x f x t



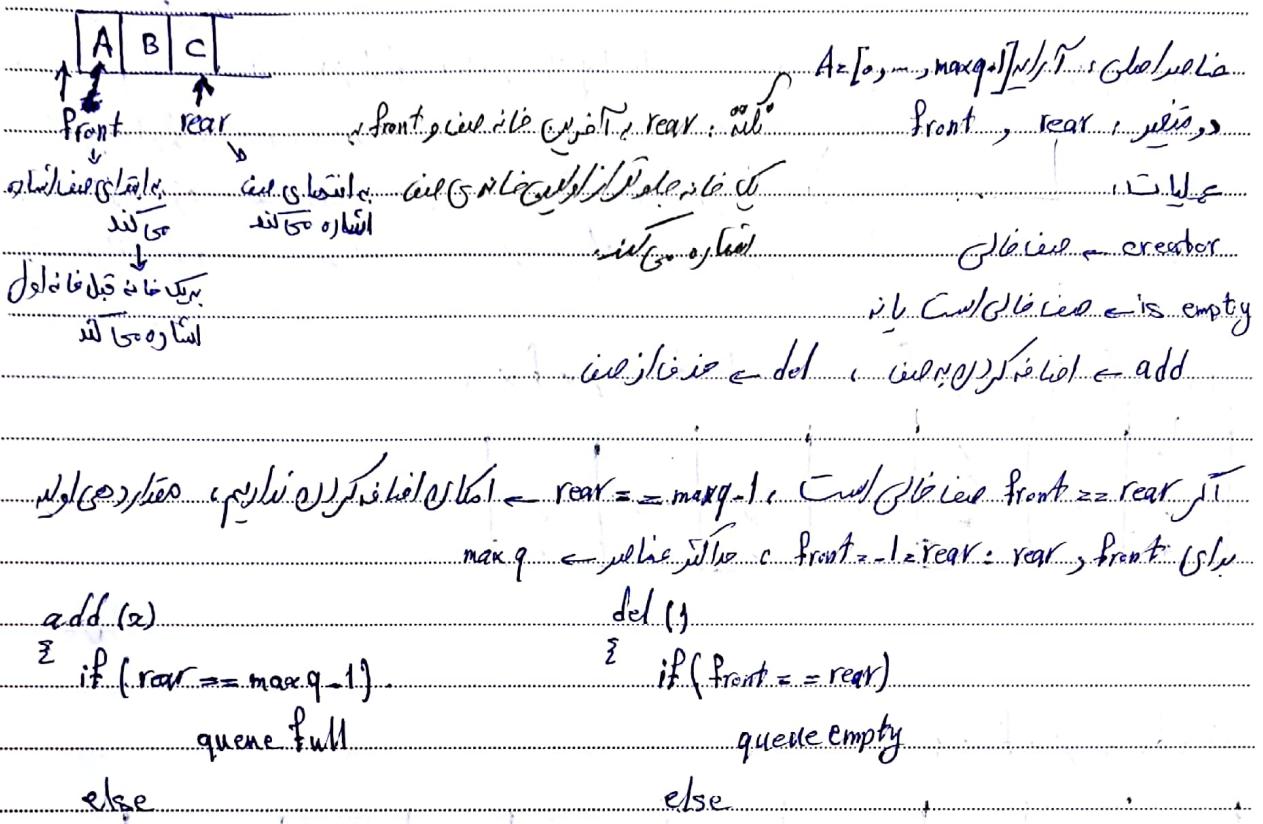
کوکیلر از اینجا (stock) بروی (stock) یا اینجا (stock) بروی (stock) برداشت می‌کنند.

Subject :

Date .....

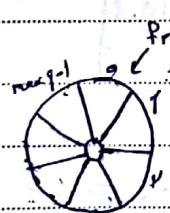
FIFO queue & circ

مقدمة في الالغاز و حلها و ازالة الالغاز حل الالغاز



$\Sigma \quad A[\text{front}] = x$

return A[front]



5

فرانت يشير إلى العنصر الأول في المجموعة و rear يشير إلى العنصر الأخير في المجموعة

maxq-1 ← العنصر الاخير front = rear ← العنصر الاول  
 $(\text{rear}+1) \bmod (\text{maxq}) = \text{front}$  ← العنصر الاول front = rear ← العنصر الاول

$$\begin{aligned} F-R &= \text{المدخلات} - F_R \\ R-F &= \text{الخروقات} - R_F \end{aligned} \quad \left. \begin{array}{l} F_R = \text{المدخلات} \\ R_F = \text{الخروقات} \end{array} \right\}$$

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

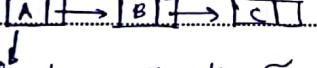
front = (front + 1) mod max\_q

return A[front]

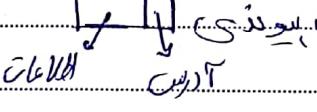
5

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)

struct node {  
 int info;  
 node\* next;  
};

  
first → A (أول)  
first → B (ثاني)  
first → null (ثالث)  
next → null (رابع)

node\* first, q;

new(first);

first → info = A;

first → next = null;

p = first;

first = first → next;

free(p);

q = first;

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

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 خطوة: node \*p

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

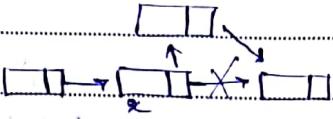
Subject :

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)

else: push(x) = insert node at top of stack & update current level to one less than previous

push(x) = insert node at top of stack & update current level to one less than previous

push(var) {

new(p)

p.pop(top) {

p.info = var if top == null

p.next = top

p = top

top = p

top = top.next

print(p.info)

free(p)

(6) (5) (4) (3) (2) (1) (0) (tail), rear = first, first = null, level = 0

if front == null, rear == first, first = null, level = 0

add(first, var) {

new(p)

else

p.info = var

rear.next = p

if front == null

front = p

Subject :

Date .....

del(front, rear) {

if front == null

p = ~~front~~ front

front = p → front → next

if front == null

rear = null

print(p → info)

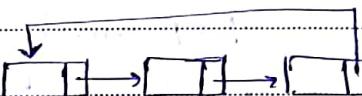
free(p)

ادخل العقد المطلوب

xx' + xz + y



اذهب إلى العقد



ادخل العقد المطلوب

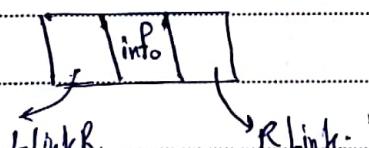
ومن أول العقد

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 .....

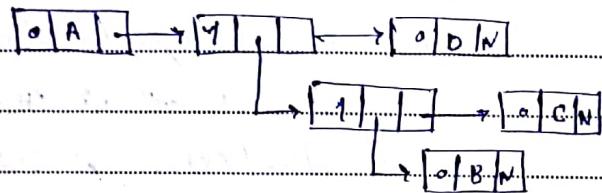


دليان (linked list)



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)

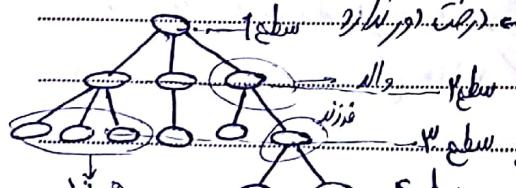
}

جذب (linked) Tree

جذب غرافي

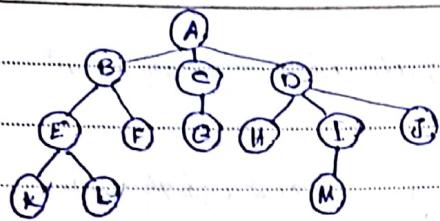
جذب لاسلكي (Wireless) Tree

جذب زيراني

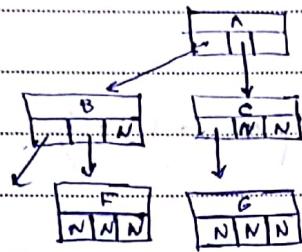


جذب بابا

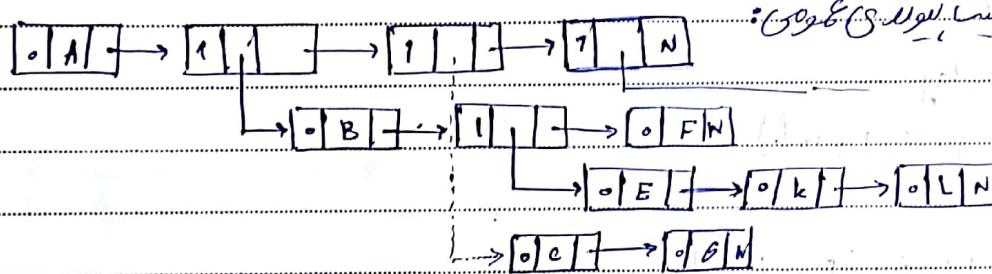
جذب ملائكة (Angel's) Tree



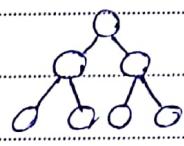
(A(B(E(k,l),F),C(G),D(H,M),I,J))



بيان ساري بالشكل ليست  
درخت دقيق بالشكل ليس دقيق بل هو دقيق  
لذلك  $\text{End}(n)$   $\text{End}(n-1)$   $\text{End}(n)$   $\text{End}(n-1)$   $\text{End}(n)$   
 $\text{End}(n-1)$   $\text{End}(n)$   $\text{End}(n-1)$

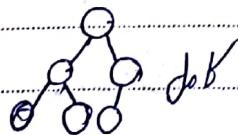


درخت دو دوی (binary) هي ترتيل تبعي بالشكل عالي و بروابط غير زير درفته جيد و غير المتساكن دار (عالي والكرد)  
فرزند دار (غير كردار)



درخت بـ ٢  
عالي كردار

درخت بـ ١، درفته كـ اگرها (رسوت جب تكون ليست في قرار كرفته بالشكل  
وبه خبر سلم كـ خربقى سلم ها با الـ كامل بالشكل



در درخت دو دوين عالي كردار (نودها) ما موجود در سلم ام =  $i-1$

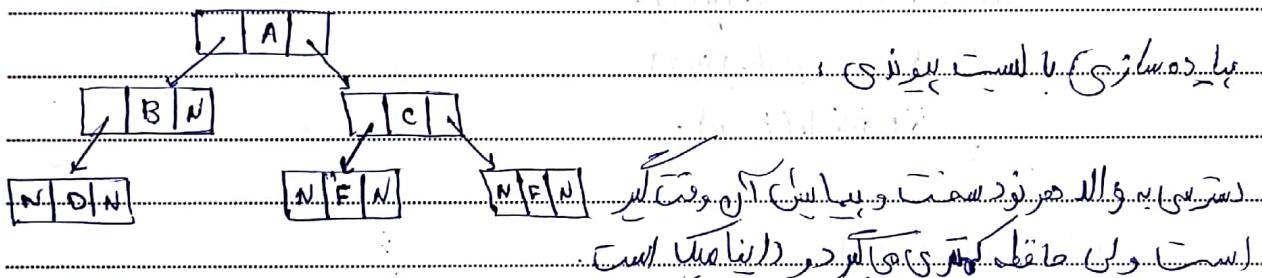
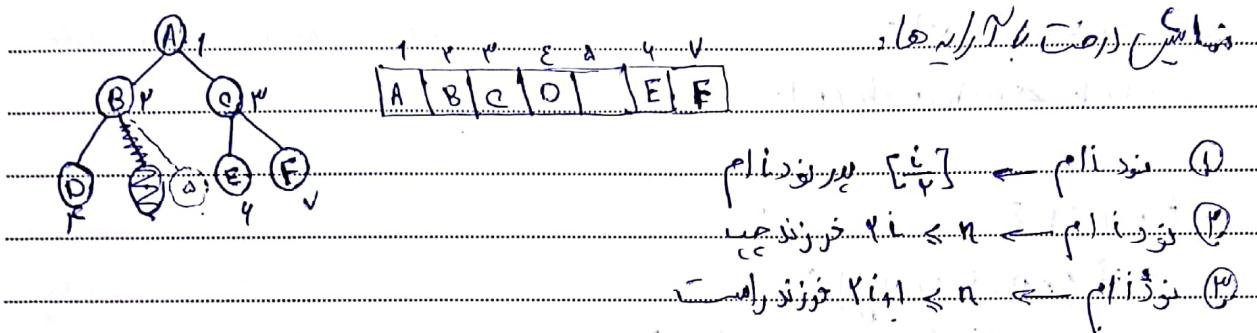
Subject :

Date .....

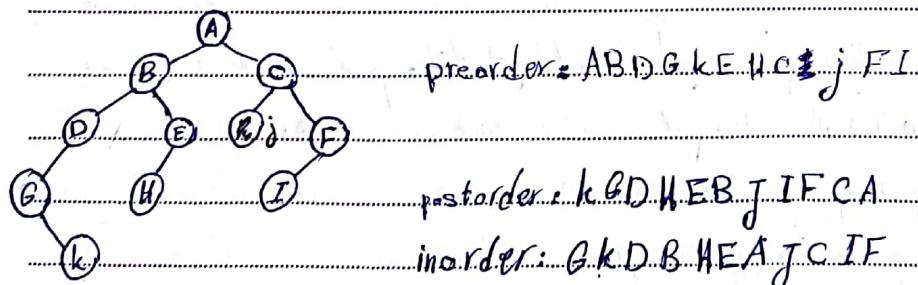
$$1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}$$

درخت باینری بروز رسانی با کمینه

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



LRV  $\leftarrow$  postorder LVR  $\leftarrow$  inorder VLR  $\leftarrow$  preorder



جواب: inorder: G, D, B, H, E, A, J, C, I, F  
postorder: k, B, D, H, I, E, B, J, I, F, C, A

الكلمة الأولى preorder، posterior (فوق)، postorder، preorder  
[أمامي]، [خلفي]، [فوق]، [فوق]، [خلفي]، [أمامي]، [خلفي]، [فوق]، [فوق]، [خلفي]، [أمامي]

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

void push(int v) {

stack.push(v);

```
int pop() {
```

if (pp.isEmpty) {

while (! pp.isEmpty()) {

`pp push (temp);`

```
return p.p.pop(1);
```

truth is also a lie, because it is not true. (وَالْحَقُّ كَاذِبٌ لَا يَرْجُعُ عَنْ حَقٍّ وَالْكَاذِبُ بِالْحَقِّ يَرْجُعُ)

مصحح کریما و مکالمہ مسیحیوں کے لئے

الآن، سأقدم لك بعض الأمثلة على تطبيق هذه المفاهيم.

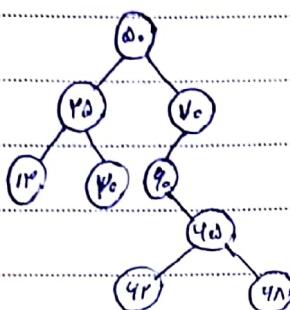
لطفاً <sup>(1)</sup> stack خود می کنیم و از آنها آنرا وارد آن می کنیم <sup>(2)</sup> اگر حالی بعد آن بضرر را وارد آنکه حفظ کنیم  
آنکه <sup>(3)</sup> stack خود با بالاترین عیار <sup>(4)</sup> کنیم فرآور برای بود نه کوچک را به stack اضافه می کنیم و  
آخر برای بود نه با آن روند <sup>(5)</sup> stack و آن عدد را حذف می کنیم (برایان <sup>(6)</sup> عددی که باقی می باشد) stack  
همان عددی <sup>(7)</sup> که از آنها بحذف شده است و آن را بعده <sup>(8)</sup> و جمعیت عددی <sup>(9)</sup> و جرد <sup>(10)</sup> و جرد <sup>(11)</sup> دارد.



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

برگشتی:

BST-search(p, key)

if (p == null)

return ("not found")

if (p->info == key)

return (p);

if (p->info > key)

return (BST-search(p->L\_Link(key)))

return (BST-search(p->R\_Link(key)))

5

جستجوی اضافی

BST-insert(p, key)

q = null

p = root

while (p != null)

if (p->info == key)

return (false)

if (p->info > key)

p = p->L\_Link

else

p = p->R\_Link

Subject :

Date .....

new(x)

جذر اهم اعداد

x.info = key

باقی قرار دهم

x.R.Link = null



x.L.Link = null



if (root == null)

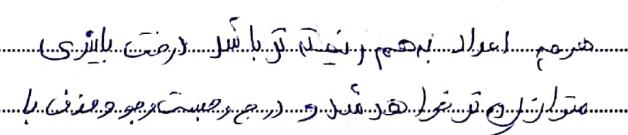


root = x

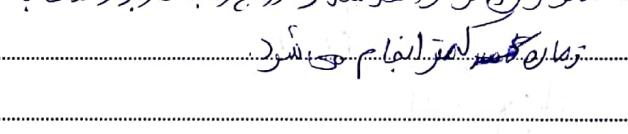


else

if (q.info > key)



q.L.Link = x



else

q.R.Link = x



return True

5

مسخر

برای جذب مسخری با هم فریاد کنند

و حقیقی عفونی با در فرزند را جذب کنند

BST.Del (root, x)

برای حذف یک عقد از درخت باید آن را پیدا کرد

if (x.R.Link == null

یا جایگزین آن را پیدا کرد

or x.L.Link == null)

وارد ساختی را بسته کرد

q = null

باید تغییر عود در اینجا باشد

p = root

لطفاً

while (p <= x)

q = p

if (p.info < x.info)

p = p.R.Link

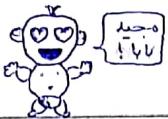
else

MICRO

p = p.L.Link

5

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 (y ≠ null)

else {

if p.info > q.info

q → R.link = y

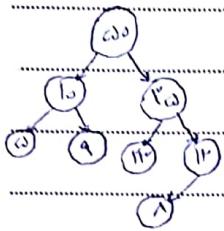
else

q → L.link = y

if p.info <

x.info < p.info

}



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

اندھیں لیہ جو جگہ سے بھی نہ مکمل ہے : Min Tree

یہی جس پر کوئی محدودیت نہیں

اندھیں جو جگہ سے بھی مکمل ہے : Maxheap

(جیسے h = logn) اندھیں جو جگہ سے بھی مکمل ہے : Min Tree h = Minheap

$O(bgn)$  : جیسے جو جگہ

اندھیں جو جگہ سے بھی مکمل ہے : Maxheap جسے جو جگہ سے بھی مکمل ہے : Minheap

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 جسے جو جگہ سے بھی مکمل ہے اسے جو جگہ سے بھی مکمل ہے اسے جو جگہ سے بھی مکمل ہے اسے

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

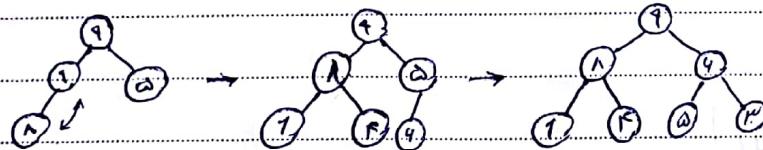
5 j = i + 1

$H[i] = k$

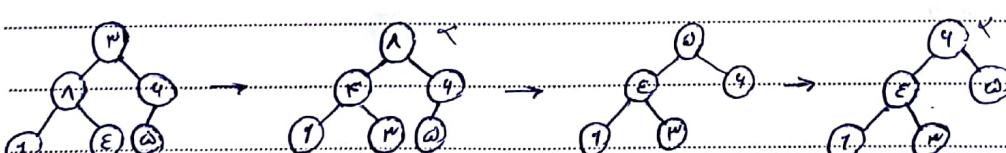
return(key)

9 1 0 A E 4 W

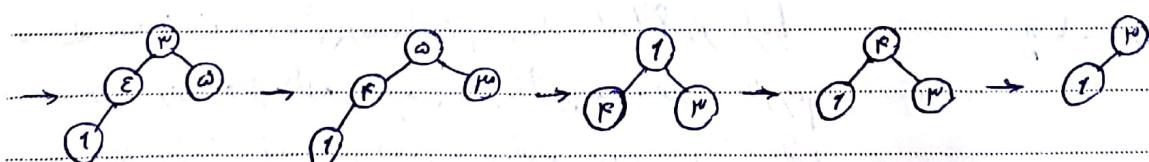
max heap (مکرر) و میکرر (معنی دارد)



21 ①



22 ①



23 ①

Subject :

Date .....

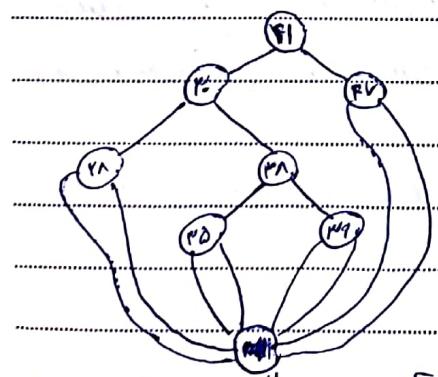
for  $i = 1$  to  $n$ max-heap-insertion ( $A[i]$ )  $\rightarrow O(n \log n)$  $\} \rightarrow O(n \log n)$ for  $i = 1$  to  $n$  $A[i] = \text{max-heap-del}(1) \rightarrow O(n \log n)$ النحوتة  $O(n \log n) \rightarrow \text{heap sort}$  ②النحوتة  $O(n \log n) \rightarrow \text{AVL tree} + \text{Inorder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

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

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

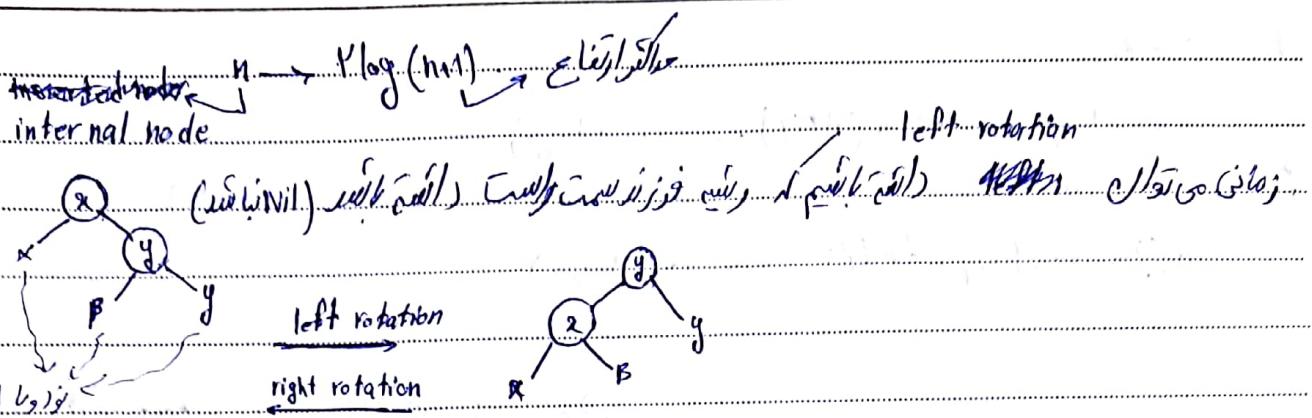
النحوتة  $O(n \log n)$  دليل المتوازن بورنج تفليس  $\rightarrow$  AVL treeblack  $\leftarrow$  null, black ① black  $\leftarrow$  root ①دليل متوازن بورنج تفليس  $\rightarrow$  AVL tree

جذر دليل متوازن بورنج تفليس

جذر دليل متوازن بورنج تفليس  $\rightarrow$  AVL treeMICRO  $\rightarrow$  Null  $\rightarrow$   $\text{Cnd/BH} = Y$  بول فـ 1  $\rightarrow$  BH  $\leftarrow$  null  $\rightarrow$  null  $\rightarrow$  black height

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 ( $T, z$ )

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

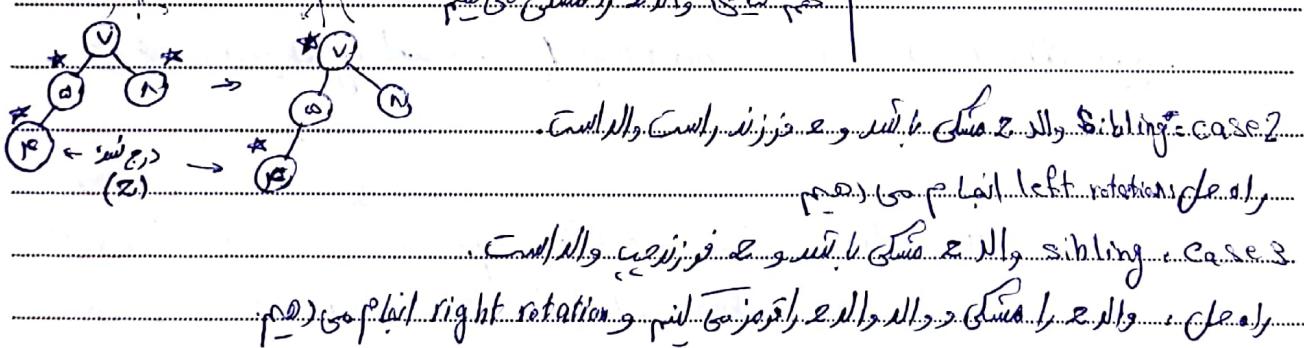
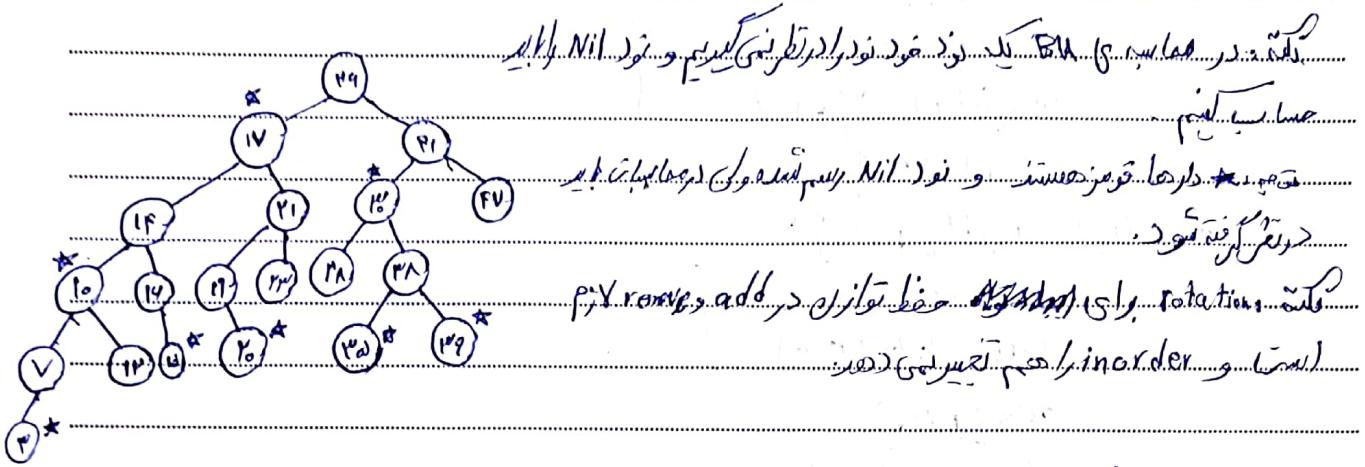
$p[z] \leftarrow \text{parent}$

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

**Subject :**

Date .....

R.B. insection line (r, z)



R.B.-insert & fix up (T & 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 of  $\text{Fe}^{7+}$  = black

Color: #47z black

Color  $\lceil p[p[x]] \rceil$  = red

else

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

~~else if (z == right[p[z]]) {~~

$$E = p[z]$$

left-natale ( $T, z$ )

~~class of 1923~~

Color  $\text{Sp}\{2\}7$  = black

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

right-rotation( $T, p[p[x]]$ )

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 خنفی عورمهز باید است: قانونی  
نفف (ب) فوج خنف میگیرد: قوانین black boot و توالی درون عورمهز و کسانی را بطور  
تعادل دارند و سکوت نمیکنند: قانونی

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)

*ARMLiPP*

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

■ 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:

inorder traversal is index of node . Rank of node is rank of node in sorted array .

as rank(T,x) {

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

y - x

while ( y ≠ root[T] )

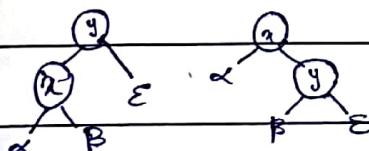
~~if (y = right [p[y]])~~

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

$$y \leftarrow p[y]$$

return r

الآن الـ  $\alpha$  هو كحد أقصى (أدنى) في  $T$ ، فـ  $\alpha$  هو جزء من  $T$ .  
نـ  $\alpha$  هو جزء من  $T$ ، فـ  $\alpha$  هو جزء من  $T$ .



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

الآن  $\alpha(\log n)$  ، rank و size (النقطة التي تصل إلى) تأخذ وقت  $\log n$  ، وبهذا فإن order static .

■ 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

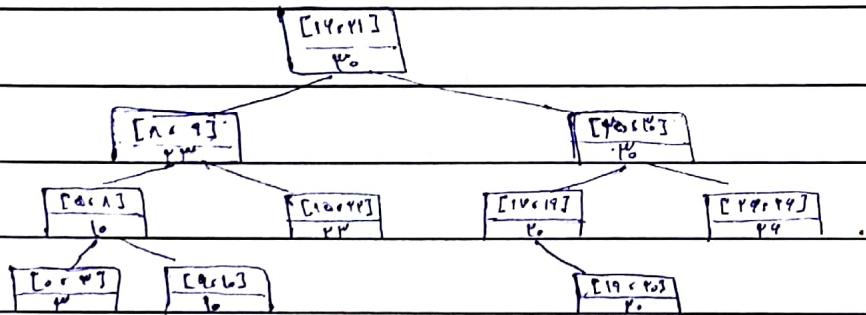
از هر دوی ایجاد

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

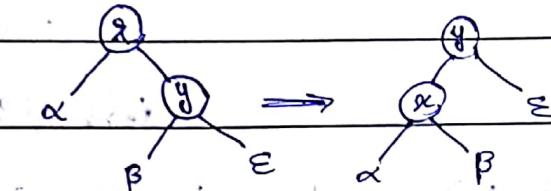
$\text{max}[x] \leftarrow \text{max}[\text{max}[\text{left}[x]], \text{max}[\text{right}[x]], \text{high}[\text{int}[x]]]$



## Subject:

Date:

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



: (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

ج) حمراء (red bracket frog). ينتمي إلى فصيلة البروبيديات (Batrachidae)، وهي فصيلة تضم العديد من الأنواع، مثل العقرباء والبروبيات. تم العثور على هذه الصور في إقليم كاراباخ، حيث تم التقاطها في منطقة بيت لاهيا.

■ Subject:

■ Date:

$i \sim i'$  : overlap between intervals  $i$  and  $i'$  if  $\max(\text{low}[i], \text{low}[i']) \leq \min(\text{high}[i], \text{high}[i'])$

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

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

interval-search( $T \times i$ ) {

$x \leftarrow \text{root}(T)$

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

        if ( $\text{left}[x] \neq \text{nil}[T]$  and  $\text{low}[i] < \max(\text{left}[x])$ )

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

        else

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

    return  $x$

§

graph  $G = (V, E)$   $V$  is a set of vertices and  $E$  is a set of edges

multiple edges between two vertices  $a$  and  $b$   $(a, b) \in E$   $\Rightarrow$   $a \sim b$

multigraph  $\Rightarrow$   $\exists (a, b) \in E$  such that  $a \sim b$

multigraph  $\Rightarrow$   $\exists (a, b) \in E$  such that  $a \sim b$

$n(n-1)/2$  edges and  $n(n-1)$  vertices

ARMS

Subject:

Date:

مختصر ملخص درس نظریه گراف

گراف از دو مجموعه متمم و مجموعه همسایه است

گراف متصدی (Connected) گرافی است که همه گره ها با یکدیگر متصدی باشند

گراف متصدی قوی (Fully Connected) گرافی است که همه گره ها با هم متصدی باشند

گراف متصدی ضعیف (Weakly Connected) گرافی است که همه گره ها با یکدیگر متصدی باشند اما ممکن است گره هایی باشند که در یک دیگر متصدی نباشند

گراف متصدی موقت (Temporary Connected) گرافی است که همه گره ها با یکدیگر متصدی باشند اما ممکن است گره هایی باشند که در یک دیگر متصدی نباشند

گراف متصدی موقت (Fully Connected) گرافی است که همه گره ها با یکدیگر متصدی باشند اما ممکن است گره هایی باشند که در یک دیگر متصدی نباشند

معادله درجه گره  $D_i = |E| \in \sum_{i=1}^n d_i$

برای محاسبه درجه گره  $D_i$  از درجه های مجاور آن محاسبه می شود

برای محاسبه درجه گره  $D_i$  از درجه های مجاور آن محاسبه می شود

برای محاسبه درجه گره  $D_i$  از درجه های مجاور آن محاسبه می شود

برای محاسبه درجه گره  $D_i$  از درجه های مجاور آن محاسبه می شود

$\sum_i a_{ui} = \deg(u)$

$\sum_i a_{vu} = \deg(v) \rightarrow \sum_i a_{ui} + \sum_i a_{vi} = 2|E|$

برای محاسبه درجه گره  $D_i$  از درجه های مجاور آن محاسبه می شود

Subject:

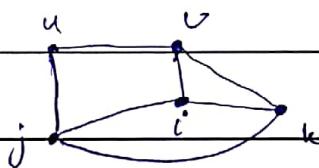
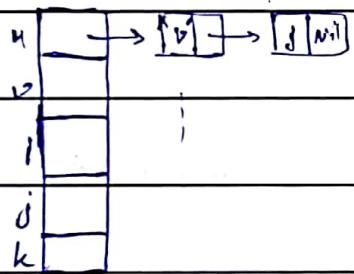
Date:

مقدمة طرق مترافقه لحل معادلات ديناميكية (الاستقر)

١) مفهوم المعرفة ٢) تطبيقات المعرفة ٣) اثباتات المعرفة

خوبی آن را باید سه تا آن در (۱) و مجموع آنها برابر با (۱) نوشته باشد.

لهم اهدنا لجهة رحمة ونورك وارزقنا ايمان يعيننا في مواجهة شرور عالمك الظاهر والباطن



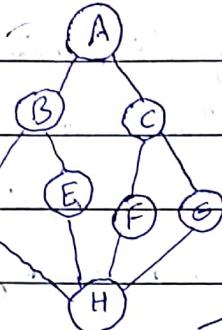
فیبرومیزایس کاربونیت دار دلستیت نوکارن از آن نوکارن فردا نیل را زیرم (مالک طبقه خود) فرآوری کسری

۲- در اینجا می‌توانید مکالمه‌ای را که با همراه خود داشتید در یک فایل صوتی ذخیره کنید و آنرا پس از مدتی مجدداً بشنید و آنرا بررسی کنید.

Preorder جملہ : (D-FS) depth first search جنگلیں ① : جنگلیں  
Inorder جملہ : (B-FS) breath first search جنگلیں ②

Subject:

Date:



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

١: سعى  
نقطة شروع للخوارزمية (بخطار دخت)

الخطوات (أ) ترتيب المنشئات في قائم وبيان ازدواجيتها، (ب) معاينة وبيان حالتها، (ج) إضافة المنشئات التي لم يمسسها إلى قائم المنشئات التي تم معاينتها.

code:

For i=1 to n :

Visited[i] = 0;

Time: O(n)

DFS(G, V) {

O(n<sup>n</sup>) أكبر معاينة ممكنة

visited[V] = 1;

for each u adjacent to V do

if Visited[u] = 0;

DFS(G, u)

}

A, B, C, D, E, F, G, H

: الـ ٦ خطوة : خطوة ١

step 1: (A) أكبر معاينة ممكنة

step 2: (B) أكبر معاينة ممكنة

Code:

For i=1 to n

Visited[i]=0;

BFS(G, V).{

Visited[V]=1;

- Add queue(V)

while(!IsEmpty(q)){

    v = del\_queue(q)

    for each i adjacent to v.

        if Visited[i]=0;

            Visited[i]=1;

            Add\_queue(q, i)

جواب این سوالیست که آنکه چندین مجموعه می باشند

3 num=0

for i=1 to n

    if Visited[i]==false

        DFS/BFS(G, i)

        num++;

آنکه چندین مجموعه می باشند

شاید DFS, BFS

یعنی چندین مجموعه می باشند

لیکن

Subject:

Date:

دخته جوی گراف G با مجموعه محدود Spanning Tree

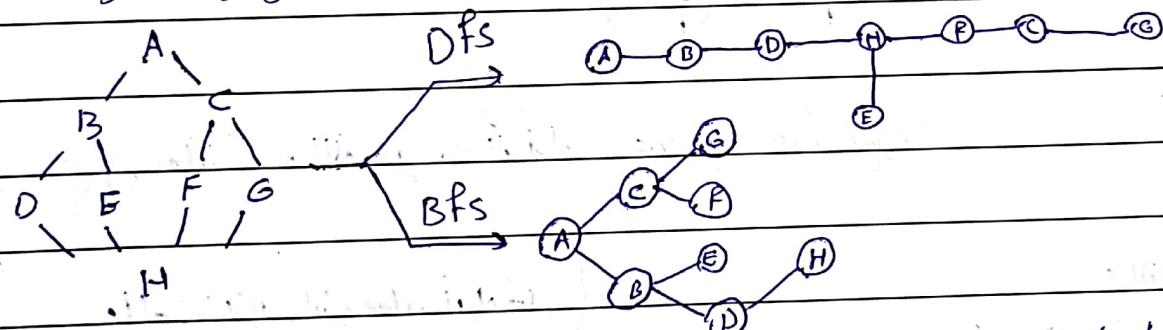
نورهای آن و نالهای E جزو تعریفی اینم که

نضج است از دلایل آن نیز مجموعه E و نورهای آن نیز مجموعه E باشد.

$E' \subseteq E$ ,  $V = V(G)$ ,  $|E'| = n - 1$  و بایدیک دخته شود!

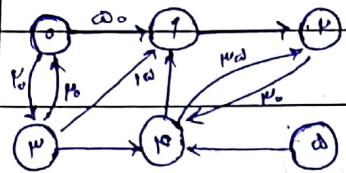
\* در نظر گیری کنید که یا زیرگرافی کافی نباشد و ممکن است بازگشتی داشته باشد  
که این نورهایی را که این داشته باشند (دخته) نویسند اما باشند

step 1:  $\leftarrow$  هر یکی از این مجموعه های ممکن را در نظر بگیریم و ممکن است داشته باشد  
که نوشتاری ممکن



Subject:

Date:



تكراف وزراعة طرفة  
حركة در تكراف على غير طرفة ويزو على طرفة  
كل طرفة تو زرها متساوية واثرها على طرفة

(، لكي تكنولوجيا قابلة للاستفادة (الطبقة الأولى)، حيث إنها (الطبقة الأولى) هي التي تمتلك

ـ حركة سهلة جداً على طرفة وجزء من طرفة

ـ كل طرفة لها اتجاه واحد (غير تكراف) اتجاه (غير تكراف)

ـ كل طرفة تو زرها متساوية

Dijkstra-shortestPath( $n, v$ ) {  
    source}

    for ( $i = 0, i < n, i++$ ) {  
        {  
             $s[i] = \text{False};$   
             $dist[i] = \text{length}[v][i]$   
            {  
                 $\downarrow$  مراجعة مجاورة  
                 $s[v] = \text{True}$   
                 $\exists dist[v] = 0$   
                for ( $i = 0, i < n - 1, i++$ ) {  
                     $u = \text{choose}(n)$  ——————  
                     $s[u] = \text{True}$

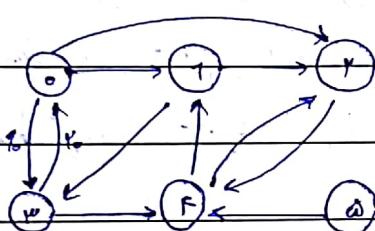
**Subject:**

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
$\alpha$	T	F	F	F	F	F

dest = [0 0 0 10 M M]

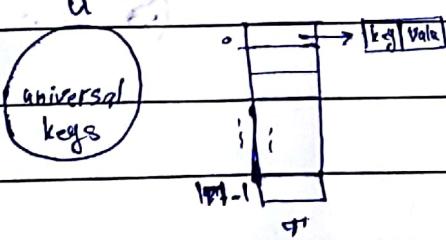
$\downarrow$   $s = \text{FFFTRFF}$

$\text{dist} = \text{dist}$

$$dist = \boxed{0 | Fa | Fo | lo | Yo | M}$$

$O(n^r)$  gives  $\tilde{O}(n^r)$

avl tree (AVL tree) key: (key, value). |V| = 101, and value is label T, and key is node N



A diagram showing a variable `x` pointing to a memory location. The memory location contains two boxes: one labeled `20` and another labeled `'A'`. Above the boxes, the words `key` and `value` are written, with arrows pointing from `key` to `20` and from `value` to `'A'`.

**■ Subject:**

Date:

$\text{insert}(T, x)$        $\left\{ \begin{array}{l} \text{Delete}(T, x) \\ T(\text{key}[x]) \leftarrow x \end{array} \right.$        $\rightarrow \text{search}(T, k)$   
 $T(\text{key}[x]) \leftarrow \text{null}$        $\left. \right\} \quad \text{return } T[k]$   
 سه خط اول دارای تابعیت است و سه خط دیگر نیستند

(hash table) جدول بحثي يحتوي على مدخلات موزعة في جدول مساحات متساوية، وتحتاج إلى دالة散列 hash function لـ mapping key إلى index.

linked list  $\rightarrow$  linked list, list  $\rightarrow$  list; index  $\rightarrow$  index, chaining

insert( $T, m$ )

insert  $x$  at the head of the list  $T[h(\text{key}[x])]$   $\rightarrow o(i)$

~~delete ( $\Gamma, x$ )~~

delete  $x$  from the list  $T[h(\text{key}[x])]$  } Worst case:  $O(n)$

search( $T, \alpha$ )

search an element with key in list  $T[b(h(k))]$ :  $x \in T[b(h(k))]$   $\Rightarrow$   $O(n)$

$\alpha = \frac{n}{m}$  ترکیب تابعی برای جستجو نمایند اور، که کسری کشند و درینجا  $m$  hash function داریم  
جایی  $\alpha$  load factor است که مقداری از اینها را در هر کلیک پرداخت کنیم که مقدار کلیکی که در هر کلیک انجام می‌شود برابر با  $\alpha$  باشد. این مقدار کلیکی که در هر کلیک انجام می‌شود  $O(1/\alpha)$  است. این مقدار کلیکی که در هر کلیک انجام می‌شود  $O(1/\alpha)$  است.

Subject:

Date:

multiplication, Division : hash function

Division:  $h(k) = k \bmod m \rightarrow$  when  $k^N$  >  $m$  then  $k^N \bmod m$

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

$0 < A < 1 \rightarrow kA \text{ will always } 0 \leq kA < m$

to index calculate key to get hash function  $\rightarrow$  open addressing

key to index calculate  $\rightarrow$  hash table

key to index calculate (hash table)  $\rightarrow$  index of  $T$ ,  $i \rightarrow$  insertion ( $T[i] \leftarrow k$ )

insert ( $T, k$ )  $\{$

$i \leftarrow$

index  $\rightarrow$  index,  $o$ , key,  $\rightarrow$  hash function  $\rightarrow$   $i$

repeat

$j \leftarrow h(k, i)$

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

$T[j] \leftarrow k$

return  $j$

$i++$

until  $i == m$

print ("hash table overflow")

$\}$

■ Subject:

■ Date:

search ( $T, k$ )

$i \leftarrow 0$

repeat

$j \leftarrow h(k, i)$

    if ( $T[j] == k$ )

        return  $j$

$i \leftarrow i + 1$

until  $i == m$  or  $T[j] == \text{null}$

return null

~~if  $T[j] == \text{null}$  then  $j \leftarrow j + 1$  until  $T[j] \neq \text{null}$~~

$$h(k, i) = (h'(k) + ci) \bmod m$$

linear probing (1) : hash func

$$h'(k) = k \bmod m$$

أي رقم يدخل في المصفوفة ينتمي إلى المصفوفة

فأنا ذاهب

$$h(k, i) = (h'(k) + c_1 i + c_2 i^2) \bmod m, c_1, c_2 \neq 0$$

quadratic probing (2)

$$\text{if } (h(k, 0) = h(k_{r+1}, 0)) \rightarrow h(k, i) = h(k_r, i)$$

double hashing (3)

$$h(k, i) = (h_1(k) + ih_2(k)) \bmod m$$

روابط تسلسليات لذريعة المدخلات

وهي مدعومة بـ  $h_1$  و  $h_2$  على المدى المدى

لـ  $m^2$

Subject:

Date:

probabilistic analysis: picture of this will give time complexity  
 expected time complexity  $\text{Avg}(\frac{1}{1-\alpha})$  which is  $\alpha \log \frac{1}{1-\alpha}$

best case

Average

worse case

insertionsort  $O(n)$  $O(n^2)$  $O(n^2)$ heapsort  $O(n \log n)$  $O(n \log n)$  $O(n \log n)$ Treesort  $O(n \log n)$  $O(n \log n)$  $O(n^2)$ mergesort  $O(n \log n)$  $O(n \log n)$  $O(n \log n)$ 

quicksort

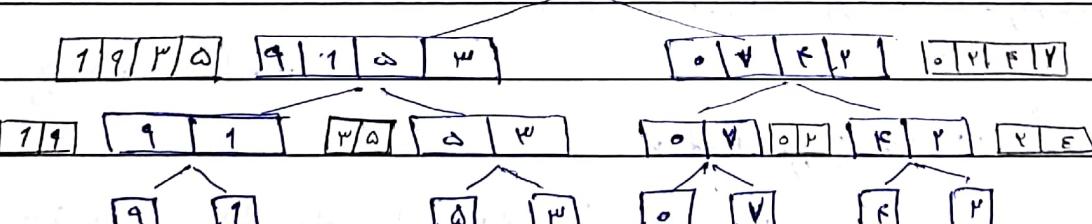
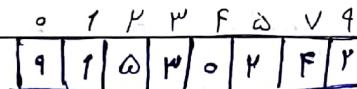
, sorting

Divide and conquer

combine ① conquer ② divide ③

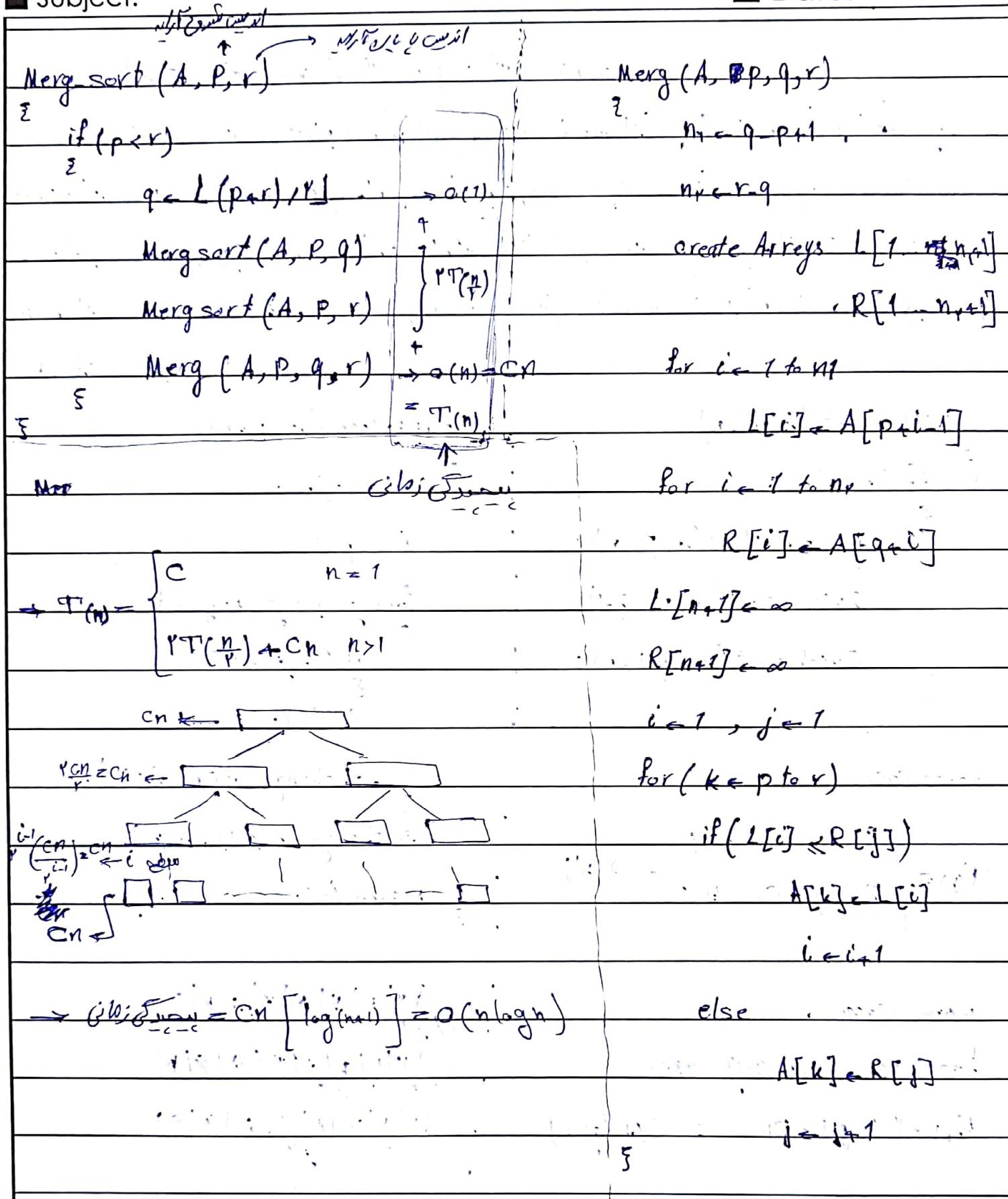
if  $n \log n$  is my base case then it's possible to get  $O(n \log n)$  solution  
 if  $\log n$  is my base case then it's possible to get  $O(n \log \log n)$  solution

mergesort



■ Subject:

■ Date:



**Subject:**

Date:

combine ⑩ conquer ⑪ divide ⑫ : Quick sort

در فنون عصر نوین راهنمایی، pivot (پیوچ) که از آن و مانند آنها برای جایگزینی و pivot (پیوچ) قرار گیرد. ولی در کاربران بسیاری از آنها که همانجا از آنها استفاده می‌کنند، pivot (پیوچ) نیز از اینهاست.

Quick-sort ( $A; p, r$ )

$\Sigma$  if ( $p \leftarrow r$ )

$i \leftarrow \text{partition}(A, p, r);$

Quick-sort( $A, p, q-1$ );

Quicksort(A, q, l, r);

worse Case:

$$T(n) = \begin{cases} c & n=1 \\ T(n-1) + cn & n>1 \end{cases} \Rightarrow O(n^r)$$

~~2018-2019~~

best case and avg cases

$$T(n) = T\left(\frac{n}{r}\right) + cn \Rightarrow O(n \log n)$$

partition(A, p, r)

$$\exists x \in A \exists r : \dots$$

$i \leftarrow p-1$

$\text{for } (j = p; j < r; j++)$

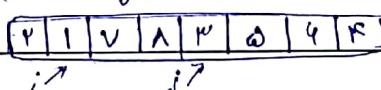
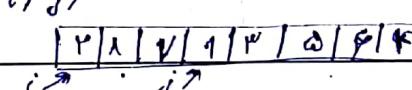
if ( $A[7] < x$ )

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

- swap exchange( $A[ij], A[eij]$ );

exchange(A[i+1], A[r]);

~~return i+1;~~



Subject:

Date:

## Dynamic programming

matrix chain multiplication problem

جامعة الملك عبد الله بن عبد الرحمن آل سعود  
جامعة الملك عبد الله بن عبد الرحمن آل سعود

$$P(n) = \begin{cases} P(k) P(n-k) & k \geq p \\ 1 & k=1 \end{cases}$$

$$\text{cost} \rightarrow C_n = \frac{1}{n+1} \binom{n}{n}$$

problem of divide and conquer divide into dynamic programming

جامعة الملك عبد الله بن عبد الرحمن آل سعود

overlap sub problems

subproblems

$A_1 A_2 A_3 \dots A_n \vdash A_i \vdash \dots \vdash A_{i,j}$

مقدمة في dynamic programming

$m[i, j] \leftarrow m[i, j]$

subproblems

$$m[i, j] = \begin{cases} 0 & i=j \\ \min_{i \leq k < j} [m[i, k] + m[k+1, j] + P_{i,k} P_{k+1,j}] & i < j \end{cases}$$

unique solution

ARMINI

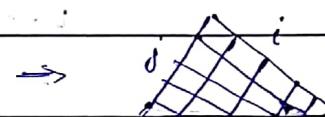
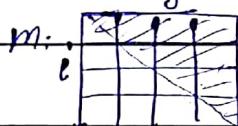
(subproblem)

Subject:

Date:

$(n \times n = \alpha(n))$  solution,  $\rightarrow$  unique solution,  $\rightarrow$  problem  $\rightarrow$   $m[i][j]$   $\rightarrow$   $s[i][j]$

matrix | Dim

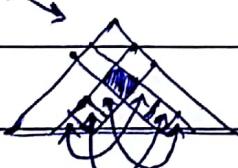
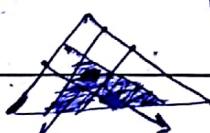
 $A_r$   $P_o \times P_d$  $A_v$   $P_d \times P_d$  $A_p$   $1 \times 1$  $A_s$   $1 \times 1$  $A_q$   $P_o \times P_d$ 

$$m[1,4] = m[1,1] + m[1,2] + m[1,3] + P_o \times 1 \times P_d = 12 \text{ vs } 0.$$

$$m[1,4] = 4448 \quad m[4,4] = 1200 \quad m[F,4] = 1000 \quad m[4,4] = 2000$$

$$\left\{ k \in P : m[1,4] + m[4,4] + P_o \times 1 \times 1 \right\}$$

$$m[1,4] = \left\{ k \in P : m[1,4] + m[4,4] + P_d \times 1 \times 1 \right\} \rightarrow S[1,4] = 12$$

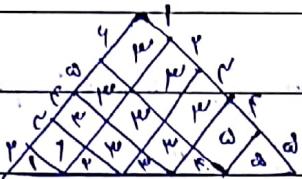
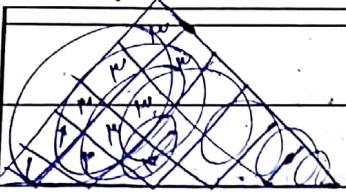


مقدار المدخلات المطلوبه من المدخلات المجهولة  $\rightarrow$   $m[1,4] = 12$

A.R.Mi.P.P

Subject:

Date:



keep a position of  $(i, j)$  till

$A_1 | A_2 | A_3 | A_4 | A_5 | A_6 | A_7 | A_8 | A_9 | A_{10} \rightarrow i=1, j=9$

$i=1, j=10 \leftarrow A_1 | A_2 | A_3 | A_4 | A_5 | A_6 | A_7 | A_8 | A_9 | A_{10} \rightarrow i=9, j=9$

$A_1 | A_2 | A_3 | A_4 | A_5 | A_6 | A_7 | A_8 | A_9 | A_{10}$

matrix-chain order ( $P$ )  $\rightarrow (P_0, P_1, P_2, \dots, P_n)$

$n = \text{length}[P] - 1$

for ( $i=1; i \leq n$ )

$m[i, j] = \infty$

for ( $j=i+1$  to  $n$ )

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

$j = i+1 \dots$

$m[i, j] = \infty$

for ( $k=i$  to  $j-1$ )

$\{$

$q = m[i, k] + m[k+1, j] + P_{j-1} P_k P_j$

if ( $q < m[i, j]$ )

$\{$

$m[i, j] = q$

$s[i, j] = k$

$\}$

$\}$

return  $m \& s$

$\}$

Subject:

Date:

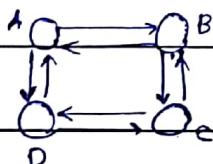
Dynamic programming problems  
optimal substructure (1)  
overlapping subproblem (1)

عوامل میکنند که مسئله را به زیر مسئله هایی تقسیم می کنند و قابل حل برای این زیر مسئله هایی هستند (1)  
و مسئله اصلی را می توان با حل زیر مسئله هایی آن را حل کرد (1)  
و مسئله اصلی را می توان با حل زیر مسئله هایی آن را حل کرد (1)  
و مسئله اصلی را می توان با حل زیر مسئله هایی آن را حل کرد (1)  
و مسئله اصلی را می توان با حل زیر مسئله هایی آن را حل کرد (1)

دو زیر مسئله داری داشت که می توانند مسئله اصلی را حل کنند (1)

shortest path ← ورک آرکیو (1)

longest path ← مسئله اصلی را حل کنند (1)



longest path ( $A \rightarrow \delta$ ): ABCD

longest path ( $A \rightarrow B$ ): ADCB

All node shortest paths

flag - workshop

ورک آرکیو (1) مسئله اصلی را حل کنند (1) و مسئله اصلی را حل کنند (1)

$$V = \{1, 2, \dots, n\}$$

$$\{1, 2, \dots, k\}$$

Subject:

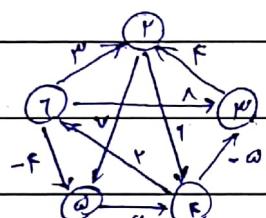
Date:

کوتاه مسیرین نازن باز که گذروند کوتاه مسیر از بیکار گذرن و برای کوتاه مسیر از بیکار گذرن

کوتاه مسیر از بیکار گذرن کوتاه مسیر از بیکار گذرن کوتاه مسیر از بیکار گذرن  
جواب بحث در ماتریس  $D$  دسته همی لینم و  $d_{ij}$  کوتاه مسیر از بیکار گذرن

$$d_{ij} = \begin{cases} w_{ij} & k=0 \\ \min \{ d_{ij}^{k-1}, d_{ik} + d_{kj} \} & k \geq 1 \end{cases}$$

حالات مجاور



$$D^0 = \begin{bmatrix} 0 & v & w & \infty & -f \\ \infty & 0 & \infty & 1 & v \\ \infty & f & 0 & \infty & \infty \\ v & \infty & -w & q & \infty \\ \infty & \infty & \infty & q & 0 \end{bmatrix} \quad D^1 = \begin{bmatrix} 0 & v & w & \infty & -v \\ \infty & 0 & \infty & 1 & v \\ \infty & f & 0 & \infty & \infty \\ v & \infty & -w & 0 & -v \\ \infty & \infty & \infty & q & 0 \end{bmatrix}$$

$$D^2 = \begin{bmatrix} 0 & v & w & f & -f \\ \infty & 0 & \infty & 1 & v \\ \infty & f & 0 & \infty & u \\ v & \infty & -w & 0 & -v \\ \infty & \infty & \infty & q & 0 \end{bmatrix} = D^3 = \begin{bmatrix} 0 & v & w & \infty & -v \\ v & 0 & -f & 1 & -1 \\ v & f & 0 & \infty & w \\ v & -1 & -w & 0 & -v \\ 1 & 0 & 1 & q & 0 \end{bmatrix}$$

با تعریف مسیر از  $i$  به  $j$  کوتاه مسیر از  $i$  به  $j$  مجاور و از نوچهای  $i$  به  $j$  مجاور و از نوچهای  $i$  به  $j$  فعلاً بود

$$\pi_{ij}^0 = \begin{cases} \text{nil} & i=j \text{ or } w_{ij} = \infty \\ i & i \neq j \text{ and } w_{ij} \neq \infty \end{cases}$$

$$\pi_{ij}^k = \begin{cases} \pi_{ij}^{k-1} & d_{ij} \leq d_{ik} + d_{kj} \\ \pi_{kj}^{k-1} & d_{ij} > d_{ik} + d_{kj} \end{cases}$$

Subject:

Date:

~~for all  $i, j$  such that  $\pi_{ij} \neq \infty$ , update  $d_{ij}$  and  $\pi_{ij}$~~

<del>Initial state</del>	$O(n^2)$	$O(n^2)$
BFS	$O(mn)$	$O(n^2m)$
DFS	$O(mn)$	$O(n^2m)$
Linear	$O(n^2)$	

Floyd-Warshall ( $w$ )

$$\Sigma D^0 = w$$

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

  for ( $j=1$  to  $n$ )

    if ( $i=j$  or  $w_{ij} = \infty$ )

$\pi_{ij} = \text{nil}$

    else {

$$\pi_{ij} = i$$

    }

  }

  for ( $i=1$  to  $n$ )

    for ( $j=1$  to  $n$ )

$d_{ij}^k = \min(d_{ij}, d_{ik}^{k-1} + d_{kj}^{k-1})$ ;

ARMISS if ( $d_{ij} < d_{ik}^{k-1} + d_{kj}^{k-1}$ ) {  $\pi_{ij}^k = \pi_{ij}^{k-1}$  }  
  else {  $\pi_{ij}^k = \pi_{kj}^{k-1}$  }

Subject :

Date .....

$$D = \begin{bmatrix} 0 & 1 & -1 & 1 & -1 \\ 1 & 0 & -1 & 1 & -1 \\ -1 & 1 & 0 & 1 & 1 \\ 1 & -1 & 1 & 0 & -1 \\ -1 & 1 & -1 & 0 & 1 \end{bmatrix} \quad D^{\omega} = \begin{bmatrix} N & W & E & S & 1 \\ E & N & W & S & 1 \\ W & S & N & E & 1 \\ S & E & N & W & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

مهمة 1: ما هي المatrice المقابلة لـ  $D^{\omega}$ ؟

الإجابة: المatrice المقابلة لـ  $D^{\omega}$  هي  $D^{\omega T}$ .

المطلب 2: ما هي المatrice المقابلة لـ  $D^{\omega}$ ؟

الإجابة: المatrice المقابلة لـ  $D^{\omega}$  هي  $D^{\omega T}$ .

$$G(V, E) \rightarrow G'(V, E')$$

$$E' = \{(i, j) \mid \text{هناك مسار من } i \text{ إلى } j \text{ في } G\}$$

$$t_{ij}^0 = \begin{cases} 1 & i=j \text{ or } (i, j) \in E \\ 0 & i \neq j \text{ and } (i, j) \notin E \end{cases}$$

$$t_{ij}^k = t_{ij}^{k-1} \text{ or } (t_{ik}^{k-1} \text{ and } t_{kj}^{k-1})$$

مهمة 3: ما هي المatrice المقابلة لـ  $t_{ij}^k$ ؟

الإجابة: المatrice المقابلة لـ  $t_{ij}^k$  هي  $T^k$ .

$$t^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

$$t_{ij}^k = t_{ij}^{k-1} \cup \{t_{ik}^{k-1} \text{ and } t_{kj}^{k-1}\}$$

$$t^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

$$t^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$t^3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$t^4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

MICRO

Subject :

Date .....

matrix multiplication

$\boxed{10}$   $A \in \mathbb{R}^{n \times n}$  و  $B \in \mathbb{R}^{n \times n}$  فی  $C = AB \in \mathbb{R}^{n \times n}$

فی  $C_{ij} = \sum_{k=1}^n A_{ik}B_{kj}$  و  $C = \begin{bmatrix} C_{11} & C_{12} & \dots & C_{1n} \\ C_{21} & C_{22} & \dots & C_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{n1} & C_{n2} & \dots & C_{nn} \end{bmatrix}$

$$AB = C$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} r & s \\ t & u \end{bmatrix}$$

$$r = ae + bh \quad s = af + bh$$

$$t = ce + dh \quad u = cf + dh$$

$$T(n) = 1 \cdot T\left(\frac{n}{2}\right) + O(n^2) = O(n^3)$$

فی  $T(n) = 1 + \frac{n}{2} + \frac{n}{4} + \dots + \frac{n}{2^k}$

جع مارکس

$$\text{کلیو: } T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n) = O(n^{\log_b a}) + \sum_{j=0}^{\log_b n} a^j f\left(\frac{n}{b^j}\right)$$

این درست کردن و مرتب کردن از قسم های mergesort.

روض با لامپ های عینک زمانی قبلی بردار (با لامپ های آغاز روی  $P_i$  را تعریف می کنیم).

$$P_i = (A_i, B_i) = (\alpha_{i1}a + \alpha_{i2}b + \alpha_{i3}c + \alpha_{i4}d)(B_i, e + B_i, f + B_i, g + B_i, h)$$

$$\alpha_{ij}, B_{ij} = \{-1, 0, 1\}$$

$$P_1 = a(f-h) = af - ah$$

$$P_2 = (a+b)h = ah + bh$$

$$P_3 = (c+d)e = ce + de$$

$$P_4 = d(g-e) = dg - de$$

Subject :

Date .....

$$P_0 = (a+d)(c+h) \quad P_1 = (b+d)(g+h)$$

مقدار فرقی بین این دو

$$P_0 = (a+d)(c+h) - ac + ah + dh \quad P_1 = (b+d)(g+h) - bg + bh + dg + dh$$

$$P_V = (a-c)(e+f) = ae + af - ce - cf$$

$$r = ae + bg = P_0 + P_4 + P_F - P_V \quad s = af + bh = P_1 + P_H$$

$$t = ce + dg = P_V + P_F \quad u = cf + dh = P_0 - P_V + P_F - P_H$$

(بلوک مولوگ)  $\leftarrow O\left(r^{\frac{1}{11}}\right)$  میز تا  $O\left(r^{\frac{\log V}{11}}\right)$  (بلوک های strassen)  $\rightarrow$

(بلوک های N.N. ترکیبی (بلوک مولوگ)  $\rightarrow$  (بلوک های غیر مترکب)  $\rightarrow$  (بلوک های غیر مترکب)

الگوریتم  $\leftarrow$  greedy

sub problem پردازش با کمترین سطح های سطحی و محدود شده top to down  $\rightarrow$  لایه هایی که هم

① optimal substructures

② greedy-choice property

اولیه ترین subproblem  $\leftarrow 1$

جواب این ترکیب را می بینیم

نسل اولیه greedy  $\rightarrow$  subproblems را می بینیم

Subject :

Date .....

: minimum spanning tree

دقيق تکلفی دارم و درخت را از اول در قدر ممکن‌ترین که از هندی، اس، ها، بلند (جبل) است.  
درخت بوسایی کوئی نیست.

$$G = (V, E) \quad G' = (V, E') \quad E' \subseteq E \quad |E'| = n-1$$

$$W(T) = \sum_{t \in T} w(t)$$

درخت با کمترین وزیر

: generic Alg

Generic-MST( $G, w$ )

=

while( $A$  does not form a spanning tree)

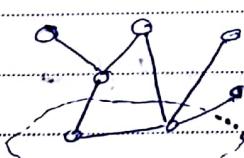
    find an edge  $(u, v)$  that is safe for  $A$

$A \leftarrow A \cup \{(u, v)\}$

return  $A$

5

ازین دلگویی پیش می‌گیرد (چنانچه اینجا  
در راستا چیزی نباشد)



تعريف خود مفهوم cut (cut(s, V-s))

cut وندیم که اینجا cut که crossing edge  
درین cut که light edge

یعنی cut که اینجا crossing edge  
ازین light edge

وهمیکی cut که اینجا crossing edge

MICRO

Subject:

Date:

• (a) Consider a  $G$  with  $n$  vertices and  $m$  edges. Let  $T'$  be a spanning tree of  $G$ . Let  $(a, b)$  be an edge in  $T'$  and  $(c, d)$  be an edge in  $G \setminus T'$ . If  $w(a, b) < w(c, d)$ , then  $T'$  is not a minimum spanning tree. To prove this, consider the spanning tree  $T''$  obtained by replacing  $(a, b)$  with  $(c, d)$ . Then  $w(T'') = w(T') - w(a, b) + w(c, d) < w(T')$ . Since  $w(c, d) \geq w(a, b)$ , we have  $w(T'') \leq w(T')$ . Therefore,  $T''$  is a minimum spanning tree, which contradicts the fact that  $T'$  is a minimum spanning tree.

• (b) Consider a graph  $G$  with  $n$  vertices and  $m$  edges. Let  $T'$  be a spanning tree of  $G$ . If  $T'$  has a cycle, then it is not a minimum spanning tree. To prove this, consider the spanning tree  $T''$  obtained by removing one edge from  $T'$ . Then  $w(T'') = w(T') - w(\text{edge removed}) < w(T')$ . Since  $T'$  has a cycle, there is at least one edge in the cycle. If we remove this edge, we still have a spanning tree  $T''$  with weight less than  $w(T')$ . Therefore,  $T''$  is a minimum spanning tree, which contradicts the fact that  $T'$  is a minimum spanning tree.

• (c) Consider a graph  $G$  with  $n$  vertices and  $m$  edges. Let  $T'$  be a spanning tree of  $G$ . If  $T'$  has a cycle, then it is not a minimum spanning tree. To prove this, consider the spanning tree  $T''$  obtained by removing one edge from  $T'$ . Then  $w(T'') = w(T') - w(\text{edge removed}) < w(T')$ . Since  $T'$  has a cycle, there is at least one edge in the cycle. If we remove this edge, we still have a spanning tree  $T''$  with weight less than  $w(T')$ . Therefore,  $T''$  is a minimum spanning tree, which contradicts the fact that  $T'$  is a minimum spanning tree.

MST.kruskal( $G, r$ )

E

$A \leftarrow \emptyset$

for each vertex  $v \in V(G)$

Make.set( $v$ )

sort the edges of  $E$  into nondecreasing order by weight

for each edge  $(u, v) \in E$ , taken in nondecreasing order by weight

if ( $\text{find-set}(u) \neq \text{find-set}(v)$ )  $\rightarrow$   $A \leftarrow A \cup \{(u, v)\}$

E

$A \leftarrow A \cup \{(u, v)\}$

union( $u, v$ )

E

return  $A$

$|E| < |V|^2 \rightarrow O(E \log V)$

$\rightarrow O(|E| \log |E|)$  for  $|E| < |V|^2$

AR.MSP  $\Rightarrow O(E \log V)$

Subject:

Date:

الخوارزمية هي إحدى الطرق التي تستخدم لحل مشكلة العرض المثلث، وهي تهدف إلى إيجاد مجموعة من الأضلاع التي تشكل مثلثاً متساوياً في المساحة.

MST Prim ( $G, r$ )

{ for each  $u \in V$  }  $\rightarrow O(V)$

$key[u] \leftarrow \infty$

$\pi[u] \leftarrow \text{null}$

$O(V \log V)$

$key[r] \leftarrow 0$

$Q \leftarrow V(G) \rightarrow Q: \text{min-priority queue} \rightarrow$  ملء قاعدة بيانات المدخلات

while ( $Q \neq \emptyset$ )  $\rightarrow O(V)$

هر نود (نقطة) له قيمة key/min

$O(\log V) \leftarrow u \leftarrow \text{Extract Min}(Q)$

if ( $u \neq r$ )

$O(VE) = O(E) \leftarrow \text{for each } v \in \text{Adj}[u]$

$A \leftarrow A \cup \{(u, \pi[u])\}$

if ( $v \in Q \wedge w(u, v) < key[v]$ )

$key[v] \leftarrow w(u, v)$

$\pi[v] \leftarrow u$

$key[v] = w(u, v)$

End /  $O(E \log V)$  بحسب الخوارزمية

الخوارزمية تهدف إلى إيجاد مجموعة من الأضلاع التي تشكل مثلثاً متساوياً في المساحة.

الخوارزمية تهدف إلى إيجاد مجموعة من الأضلاع التي تشكل مثلثاً متساوياً في المساحة.

الخوارزمية تهدف إلى إيجاد مجموعة من الأضلاع التي تشكل مثلثاً متساوياً في المساحة.

ARMISS

Subject:

Date:

8/1/20

a dynamic programming (DP)

nb state (n), update state (i) : state  $\min_i \{ \}$

nb a, a, ..., a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z

problem is called to the min value of dp[i]

$$dp[n] = dp[n-a_i] + 1$$

$$dp[n] = dp[n-a_i] + 1 \quad \left\{ \begin{array}{l} dp[n] = \min \{ dp[n-a_i] \} + 1 \\ \text{with } a_i \text{ is risk} \end{array} \right.$$

$$dp[n] = \min \{ dp[n-a_i] \} + 1$$

$$dp[a_i] = 1 \quad \text{risk}$$

select the cost  $O(nk)$  time complexity  
min of cost  $\sum dp[i]$  over all possible ways

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

update(i);

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

(O(kn))

0	1	2	3	4	5	6	7	8	9	10	11	12
0	1	2	3	4	5	6	7	8	9	10	11	12

-	1	2	3	4	5	6	7	8	9	10	11	12
0	1	2	3	4	5	6	7	8	9	10	11	12

لهم ينفعه كأداة لحل المسائل، وبه ميزة أنه يزيد من دقة حل المسائل، ويعطي إجابات صحيحة في كل الحالات، لكنه يتطلب وقتاً طويلاً (O(nk)) وبهذا نعمه في حل المسائل.

Subject:

Date:

$$dp[m] = \max [dp[m-w_i] + v_i] \rightarrow$$

$$1 \leq i \leq k$$

$$dp[m] = \max [dp[m-w_i] + v_i] \rightarrow dp[m][k] \quad (1)$$

$$dp[m-w_i][i-1] + v_i \rightarrow dp[m][k-1]$$

مقدار المكانت  $\Omega(mk)$  (البرهان)

جذر المكانت  $\sqrt{m}k$  (Dijkstra)

افضلية المكانت  $\Omega(mk)$  (البرهان)

افضلية المكانت  $\Omega(mk)$  (البرهان)

$O(\sqrt{m}k)$

### Activity selection

خواص المكانت  $\Omega(n^2)$  (البرهان)

task  $a_1, a_2, a_3, \dots, a_n$

start  $s_1, s_2, s_3, \dots, s_n$

end  $f_1, f_2, f_3, \dots, f_n$

مقدار المكانت  $\Omega(n^2)$  (البرهان)

ARMSPP  $\subseteq \{a_i, s_i, f_i\}_{i=1}^n$

Subject:

Date:

$$S_{0,n+1} = S$$

$$S_{ij} = \emptyset \rightarrow f_i \leq s_k < f_k \leq s_j \\ i \geq j \quad s_i \leq f_i$$

if two overlaid  $a_j$  &  $a_i$  have task prior  $s_j$  gives

new task  $s_{n+1}$  from  $f_{n+1}, a_{n+1} \rightarrow f_{n+1}, a_n$

### Dynamic programming

{ optimal substructure ①  
recursive solution ② }

$$A_{0,n+1} = A_{0,k} \cup \{a_k\} \cup A_{k,n+1}$$

$$|A_{0,n+1}| = |A_{0,k}| + 1 + |A_{k,n+1}|$$

$$C[i,j] = \begin{cases} 0 & S_{ij} = \emptyset \\ \max_{i \leq k < j} [C[i,k] + 1 + C[k,j]] & S_{ij} \neq \emptyset \end{cases}$$

for task  $i, j$  & subtask  $i, j$   $C[i,j]$

subproblem  $C[i,j]$  has  $C[i,k]$  &  $C[k,j]$

$C[i,k] \rightarrow j-i-1$  &  $C[k,j] \rightarrow k-j$

greedy approach

choose global task  $a_m$  &  $a_j$  from  $a_i, a_{i+1}, \dots, a_{j-1}$  &  $a_{m+1}, \dots, a_{n+1}$  &  $a_m$  &  $a_j$  have task prior  $s_j$  gives new task  $s_{n+1}$  from  $f_{n+1}, a_{n+1} \rightarrow f_{n+1}, a_m$

Subject:

Date:

$$f_m = \min_{k \in S_{ij}} (f_k + a_k \in S_{ij})$$

$S_{ij} \rightarrow$

$$S_{0,n+1} = \emptyset$$

$$S_{1,n+1}$$

$$S_{im} = \emptyset$$

$$S_{mj}$$

i	1	2	3	4	5	6	7	8	9	10	11	12
$a_1$	1	2	0	0	1	0	1	1	1	1	1	1
$a_2$	1	0	4	1	1	9	10	11	11	11	11	11

$$A_{0,n+1} = \{a_1, a_2, a_3, a_4, a_5\} \quad S_{1,n+1} = \{a_3, a_4, a_5, a_6, a_7, a_8\}$$

$$S_{1,n+1} = \{a_3, a_4, a_5\}$$

$$S_{1,n+1} = \{a_6\}$$

Recursive AS ( $s, f, i, n$ )

{ initial

: Over recursive

: while ( $m < n$  &  $s_m < f_i$ ) :

{

$m \leftarrow m + 1$

{

if ( $m < n$ ) :

{

return  $\{s_m\} \cup$  Recursive AS ( $s, f, m, n$ )

{ else

{

return  $\emptyset$

{

■ Subject:

■ Date:

greedy AS ( $s, t$ )

{  
n = len(s)

A = { $a_i$ }

i = 1

for (m = 4 to n)

{  
if ( $s_m \geq t_i$ )

A = A  $\cup$  { $a_m$ }

i = m

F  
return A;

§

\* جملہ ۱/۴ iterative پلیٹی

fractional (P) discrete (D)

کوئی کام کا وزن  $w_i$  کو کام کا مقدار  $v_i$  کے مقابلے میں اگر کام کا مقدار کو کام کا وزن کے مقابلے میں اگر کام کا مقدار کو کام کا وزن کے مقابلے میں

I = {increasing}  $v_k < w_k$

w discrete

$\sum w_k \leq w$  max  $\sum v_i$

کام کا مقدار کو کام کا وزن کے مقابلے میں اگر کام کا مقدار کو کام کا وزن کے مقابلے میں

. کام کا مقدار کو کام کا وزن کے مقابلے میں اگر کام کا مقدار کو کام کا وزن کے مقابلے میں

(P) کو (D) کا جعل کر کر کام کا مقدار کو کام کا وزن کے مقابلے میں اگر کام کا مقدار کو کام کا وزن کے مقابلے میں

Subject:

Date:

Fractional knapsack ( $w[1..n]$ ,  $v[1..n]$ ,  $w$ )  $w, v$  are sorted

{  
for ( $i = 1$  to  $n$ )

$x[i] = 0 \rightarrow$  if  $v[i] > w$  then  $x[i] = 1$

weight = 0

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

{  
if (weight +  $w[i] \leq w$ )

$x[i] = 1$

weight = weight +  $w[i]$

else

$x[i] = (w - \text{weight}) / w[i]$

weight =  $w$

break;

{  
return  $x$ ;

}

public class FractionalKnapsack {  
Dynamic Programming

objectives: Since this problem has overlapping subproblem like  $\text{find } c_i$  in  $\text{find } c_j$  and optimal substructure of previous  $c_i$ . And subproblems like  $\text{find } c_i$  in  $\text{find } c_j$  and  $\text{find } c_i$  in  $\text{find } c_k$  are related. So we can solve subproblem like  $\text{find } c_i$  in  $\text{find } c_j$  by solving  $\text{find } c_i$  in  $\text{find } c_k$  and then using  $\text{find } c_k$  in  $\text{find } c_j$ . So we can use dynamic programming to solve this problem. And optimal substructure is

(Recursive solution)

Subject:

Date:

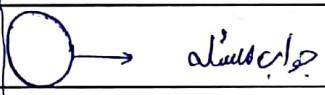
الخوارزمية المترافقه لـ knapsack problem

$$m[i, w] = \begin{cases} m[i-1, w] & w_i < w \end{cases}$$

$$m[i, w] = \begin{cases} \max(m[i-1, w], m[i-1, w-w_i] + v_i) & w_i \leq w \end{cases}$$

i	1	2	3	4	w	v
v	10	5	4	3		
w	10	5	4	3	15	20

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



الخوارزمية المترافقه لـ knapsack problem

خوارزمية knapsack problem

$$k[i, w] = \begin{cases} 1 & \text{if } w_i \leq w \text{ & } m[i-1, w-w_i] + v_i > m[i-1, w] \\ 0 & \text{otherwise} \end{cases}$$

لكل زر يعود إلى k[i, w]

$$k[n-1, w-w_n]$$

لكل زر يعود إلى k[n, w]

لكل زر يعود إلى k[n-1, w]

ARMinPP (أمثل a(nr))

■ Subject:

■ Date:

Alg. 0-1 knapsack ( $v$ ,  $w$ ,  $n$ ,  $W$ )

for ( $k = 0$  to  $W$ )

$V[i, k] = 0$

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

for ( $w' = 0$  to  $W$ )

if ( $(w[i] \leq w') \& (v[i] + V[i-1, w'-w[i]] > V[i-1, w'])$ )

$V[i, w'] = v[i] + V[i-1, w'-w[i]]$

$k[i, w'] = 1$

else

{

$V[i, w'] = V[i-1, w']$

$k[i, w'] = 0$

{ }

K, W

for ( $i = n$  down 1)

if ( $k[i, K] = 1$ )

print(i)

$O(nw)$  ~~if  $i \in \{1, 2, \dots, n\}$~~

$K = K - w[i]$

{ }

return  $V[n, W]$

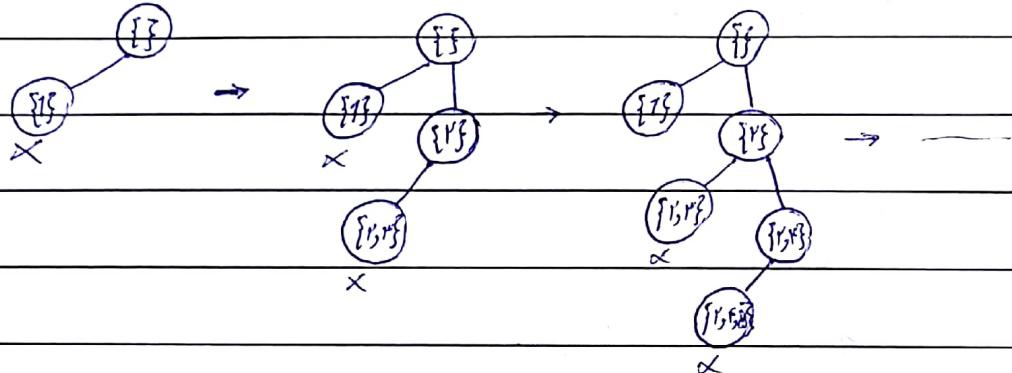
{ }

Backtracking

Initial state of tree  $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$   $\rightarrow$   $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \setminus \{1\}$   $\rightarrow$   $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \setminus \{1, 2\}$   $\rightarrow$   $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \setminus \{1, 2, 3\}$   $\rightarrow$   $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \setminus \{1, 2, 3, 4\}$   $\rightarrow$   $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \setminus \{1, 2, 3, 4, 5\}$   $\rightarrow$   $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \setminus \{1, 2, 3, 4, 5, 6\}$   $\rightarrow$   $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \setminus \{1, 2, 3, 4, 5, 6, 7\}$   $\rightarrow$   $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \setminus \{1, 2, 3, 4, 5, 6, 7, 8\}$   $\rightarrow$   $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \setminus \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$   $\rightarrow$   $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \setminus \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

■ Subject:

Date:



فرهنگ میکانیزم  $n=2$  باشد. فرضیه های این مدل این است که در  $n=2$  باید توزیع احتمالات  $\pi_{ij}$  و  $\pi_{ji}$  را بتوانیم.

مقدمة في الالكترونيات (الدوائر المتصلة) (Introduction to Electronics (Circuits))

أولاً سننظر في دورة ملء السرعات (Saturation Loop).

نحسب على المدى:

$q_1, q_r, q_p, q_f, \{1, \dots, f\}, \{1, 1, \dots, f\}, \{1, 1, 1, \dots, f\}, A, R, M, S, P$

Subject:

Date:

```
void queen(i)
{
    if(promising(i)) → إذا توفرت رانققنة فما يكتب هنا
    {
        if(i==n)
            return (c1[1] ... cn[n])
        else
            for(j=1; j<n; j++)
                col[i+1]=j
                queen(i+1)
    }
    queen(0)
```

```
bool promising(i)
{
    k=1
    switch = true
    while(k<i && switch)
    {
        if(col[i]==col[k] || abs(col[i]-col[k]) == i-k)
            switch = false;
        k++;
    }
    return switch
}
```

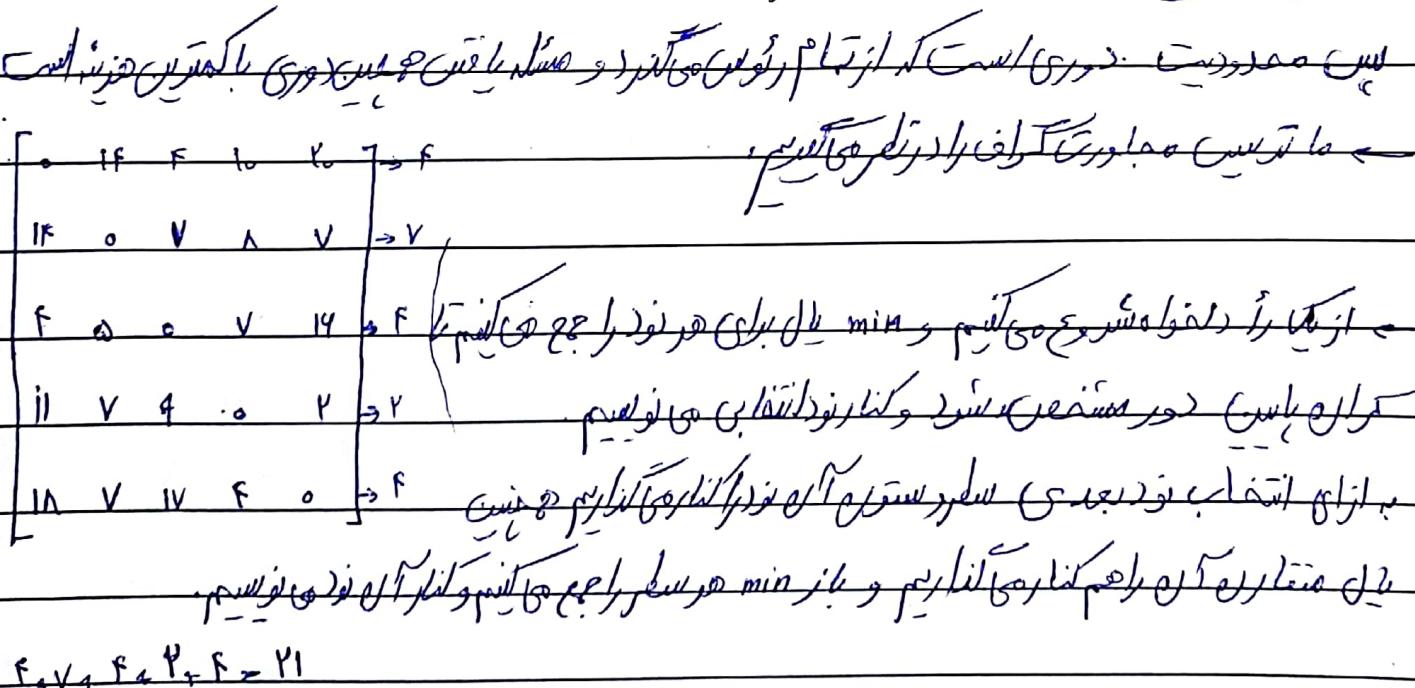
■ Subject:

■ Date:

١) back track / Branch & Bound (B&B): back track + Branch & Bound  
٢) back track (١) بحث بعمق (BFS) ثم Branch & Bound ثم BFS زاغي  
٣) branch & bound (٢) لـ local optimization (local Branch & Bound) ، (٤) local search (٤)  
٥) back track (٦) ذر وابحث (pruning)

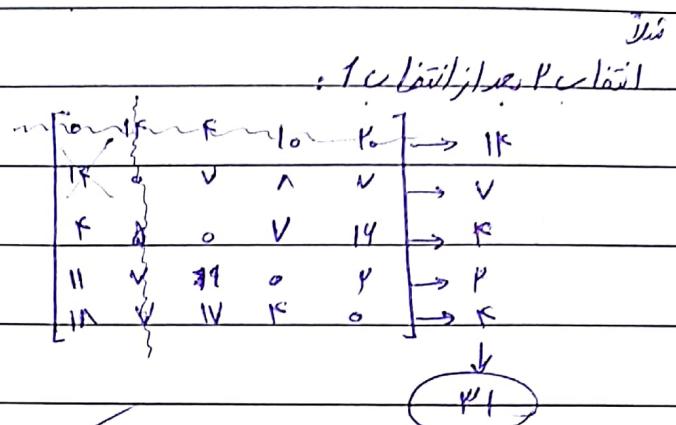
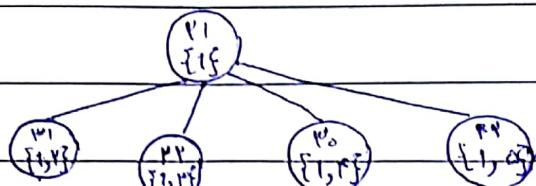
٧) max & min (٧) بـ Branch & Bound ، (٨) Branch & Bound  
٩) فروزنگي (٩)

١٠) Traveling Salesman Problem (TSP) : traveling salesman  
١١) از این میان کدام را انتخاب کنیم

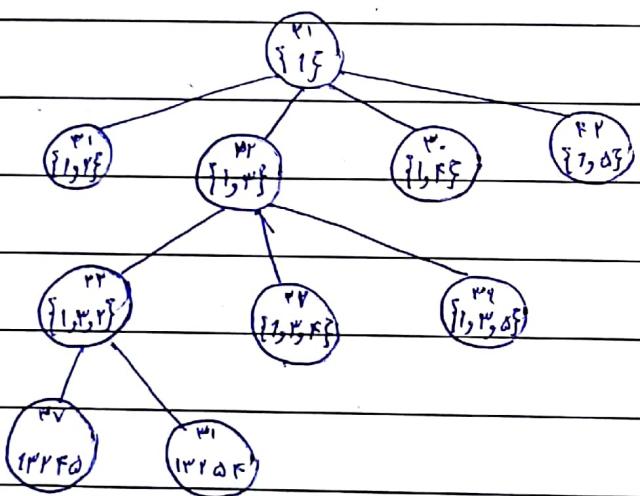


Subject:

Date:



١٥ بائیلیس پرلائی (۱۰) فوریٰ / (۱۱) سوچیلیس (۱۲) فوریٰ / (۱۳) سوچیلیس (۱۴) فوریٰ / (۱۵) سوچیلیس (۱۶) فوریٰ / (۱۷) سوچیلیس (۱۸) فوریٰ / (۱۹) سوچیلیس (۲۰) فوریٰ / (۲۱) سوچیلیس (۲۲) فوریٰ / (۲۳) سوچیلیس (۲۴) فوریٰ / (۲۵) سوچیلیس (۲۶) فوریٰ / (۲۷) سوچیلیس (۲۸) فوریٰ / (۲۹) سوچیلیس (۳۰) فوریٰ / (۳۱) سوچیلیس (۳۲) فوریٰ / (۳۳) سوچیلیس (۳۴) فوریٰ / (۳۵) سوچیلیس (۳۶) فوریٰ / (۳۷) سوچیلیس (۳۸) فوریٰ / (۳۹) سوچیلیس (۴۰) فوریٰ / (۴۱) سوچیلیس



١٥ بائیلیس پرلائی  
١٥ فوریٰ / (۱۰) سوچیلیس (۱۱) فوریٰ / (۱۲) سوچیلیس (۱۳) فوریٰ / (۱۴) سوچیلیس (۱۵) فوریٰ / (۱۶) سوچیلیس (۱۷) فوریٰ / (۱۸) سوچیلیس (۱۹) فوریٰ / (۲۰) سوچیلیس (۲۱) فوریٰ / (۲۲) سوچیلیس (۲۳) فوریٰ / (۲۴) سوچیلیس (۲۵) فوریٰ / (۲۶) سوچیلیس (۲۷) فوریٰ / (۲۸) سوچیلیس (۲۹) فوریٰ / (۳۰) سوچیلیس (۳۱) فوریٰ / (۳۲) سوچیلیس (۳۳) فوریٰ / (۳۴) سوچیلیس (۳۵) فوریٰ / (۳۶) سوچیلیس (۳۷) فوریٰ / (۳۸) سوچیلیس (۳۹) فوریٰ / (۴۰) سوچیلیس (۴۱) فوریٰ / (۴۲) سوچیلیس

(۱۰) فوریٰ / (۱۱) سوچیلیس (۱۲) فوریٰ / (۱۳) سوچیلیس (۱۴) فوریٰ / (۱۵) سوچیلیس (۱۶) فوریٰ / (۱۷) سوچیلیس (۱۸) فوریٰ / (۱۹) سوچیلیس (۲۰) فوریٰ / (۲۱) سوچیلیس (۲۲) فوریٰ / (۲۳) سوچیلیس (۲۴) فوریٰ / (۲۵) سوچیلیس (۲۶) فوریٰ / (۲۷) سوچیلیس (۲۸) فوریٰ / (۲۹) سوچیلیس (۳۰) فوریٰ / (۳۱) سوچیلیس (۳۲) فوریٰ / (۳۳) سوچیلیس (۳۴) فوریٰ / (۳۵) سوچیلیس (۳۶) فوریٰ / (۳۷) سوچیلیس (۳۸) فوریٰ / (۳۹) سوچیلیس (۴۰) فوریٰ / (۴۱) سوچیلیس (۴۲) فوریٰ / (۴۳) سوچیلیس