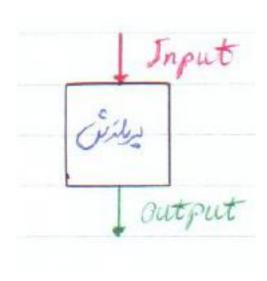
طراحى الگوريتم ها پيچيدگى الگوريتم ها

استاد درس: مهدی جبل عاملی

تعریف الگوریتم

الگوریتم یک روال محاسباتی است که مجموعه ای از مقادیر را به عنوان ورودی می گیرد و مجموعه ای از مقادیر را به عنوان خروجی بر می گرداند.



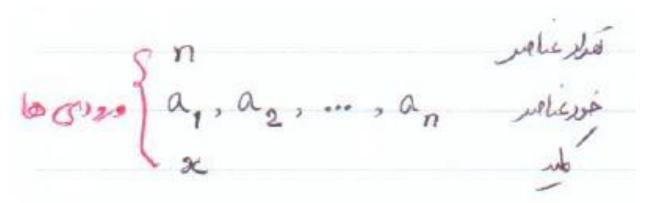
جستجو (Search)

ورودی ها:

جستجو (Search)

• ورودی ها:

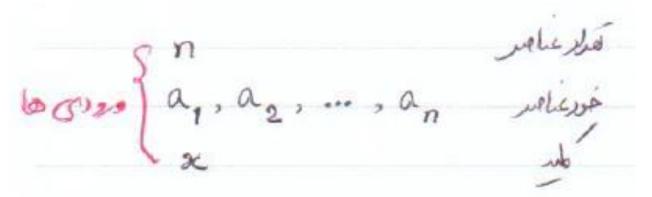
• خروجي :



جستجو (Search)

ورودی ها:

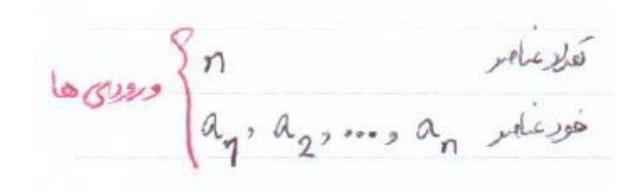
• حروجي :



وهبور در موقعی بودرس عامر کردیا کا خرای کا

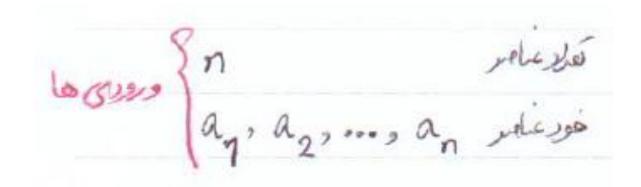
ورودی ها:

ورودی ها:



ورودی ها:

خروجي:



ورودی ها:

۰ خروجي :

olsi => a:, aiz, ..., ain

ورودی ها:

• خروجي ا

ورودی ها:

• خروجي:

Culniquistin intina

la gues o an prieses

olis => a:, aiz, ..., ain

a: {aiz < ... < ain

الگوريتم جستجوى خطى

```
index linear-search (n, A[1...n],x)}
     L=1;
   while (i kn) and (A [i] = x)
    if (c>n)
     return i;
```

```
الگوریتم جستجوی دودویی ( مر [۱۰۰۱] عند Binary-Search ( مر الگوریتم الگوریت
                                                     1=1; H=n;
                                             while (L < H)
                                          M = \left( (l + H) / 2 \right);
                                            if x = A [M] return M;
                                                                  if x < A[M] H=M-1;
                                                                                                                                                                   2 =M+1;
                                                       return 0; }
```

```
index linear-search (n, A[1...n], x)}
     while (i kn) and (A [i] = x)
     if (c/n)
         L=0;
     return i;
```

```
مقایسه دو الگوریتم جستجو
index Binary-Search (n, A[1...n],x)
    l=1; H=n;
   while (L < H)
       M=[(L+H)/2];
   if K=A[M]
                    return M;
                    H=M-1;
     if n < A[M]
        else 2 = M + 1;
    return o;
```

زمان اجراى الگوريتم (Running Time)

• تعداد دفعات اجرای عمل اصلی بر اساس اندازه ورودی برای یک نمونه(instance) از مساله.

(اگر تعداد دفعات اجرای عمل اصلی برای نمونه های یک مساله با اندازه ورودی یکسان، برابر باشد با (T(n) نمایش داده می شود)

مجموع عناصر یک آرایه

```
number Add (n, A[1...n])}
    Sum = 0;
   Sor i=1 to n do
Jum = Jum + A [i]
 return sum;
```

مجموع عناصر یک آرایه

number Add (n, A[1...n])} Sum = 0; Sor i=1 to n do Jum = Jum + A [i] return sum;

7(n) = n

كاهش زمان اجراى مساله مجموع عناصر يك آرايه (؟)

number Add (n, A[1...n])} Sum = 0; Sor i=1 to n do Jum = Jum + A [i] return sum;

كاهش زمان اجراى مساله مجموع عناصر يك آرايه

```
number Add (n, A[1...n])}
    Sum = X;
   Sor i=x to n do
Jum = Jum + A [i]
  return sum;
```

كاهش زمان اجراى مساله مجموع عناصر يك آرايه

number Add (n, A[1...n])} Sum = X: Sor i=x to n do Jum = Jum + A [i] return sum;

=> T(n) = n-7

```
number Add2 (n, A[1-n,1-n]) {
  3um = 0 ;
  For i=1 to n do
     for into n do
        Sam = Sum + A[i,j];
  return sum;
```

number Add2 (n, A[1-n,1-n]) } 3um = 0 ; For i=1 to n do for into n do Sum = sum + A[i,j]; return sum;

اندازه ورودی؟؟؟

```
number Add2 (n, A[1..n,1..n]) }
      Sum = 0;
nxn Sor i=1 to n do

e nx1 Sor j=1 to n do

e sum = sum + A[i,j];
      return sum;
```

number Add2 (n, A[1..n,1..n]) { 3um = 0; return sum;

مجموع عناصر یک آرایه دو بعدی غیر مربعی

```
number Add3 (n, m, A[1...n, 1...m7) }
     Bum = 0 ;
 For i=1 to n do

Sor j=1 to m do
               Sum = Sum + A [i]]
     return sum;
```

مجموع عناصر یک آرایه دو بعدی غیر مربعی

Bum = 0 ; For ist to a do 800 7=1 to m do Sum = Sum + A [i,]] return sum;

مجموع عناصر یک آرایه دو بعدی غیر مربعی

```
number Add3 (n, m, A[1..n, 1..m]) \\

Sum = 0;

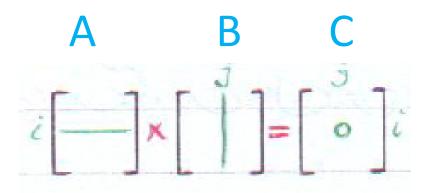
Sor i=1 to n do

Sor j=1 to m do

Sum = Sum + A[i,j] => I(n, m) = n \times m

Teturn Sum;
```

ضرب دو ماتریس



ضرب دو ماتریس

$$C[i,j] = 0;$$

$$Sor k=1 80 n do$$

- c[i,j] = c[i,j] + A[i,k] * B[k,j];

Matrix Multiply (n, A[1..n], B[1..n], 1..n]) { for i=1 to n do for 7=1 to n do c[i,j] = 0; for k=1 to n do - c[i,j] = c[i,j] + A[i,k] * B[k,j]; return C; }

ضرب دو ماتریس

A B C
$$i \left[-\frac{1}{2} \times \left[\frac{1}{2} \right] \right] = \begin{bmatrix} 0 \\ 0 \end{bmatrix} i$$

Matrix Multiply (n, A[1..n], B[1..n], 1..n]) for i=1 to n do for 3=1 to n do c[i,j] = 0; For k=1 to n do - c[i,j] = c[i,j] + A[i,k] * B[k,j]; return C; }

ضرب دو ماتریس

A B C $\begin{bmatrix} J \\ J \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} i$

عمل اصلی؟؟؟؟

Matrix Multiply (n, A[1..n], B[1..n], 1..n]) for i=1 to n do for J=1 to n do c[i,j]=0; for k=1 to n do - c[i,j] = c[i,j] + A[i,k] * B[k,j]; return C; }

ضرب دو ماتریس

 $A \qquad B \qquad C$ $i \left[-\frac{1}{2} \times \left[-\frac{1}{2} \right] = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right] i$

با فرض عمل اصلى=ضرب

Matrix Multiply (n, A[1..n], B[1..n], 1..n]) { ضرب دو ماتریس for j=1 to n do return C; } I(n)=nxnxn =n

ضرب دو ماتریس غیر مربعی

```
Matrix Multiply (n, m, p, A[1...n, 1...m], 13[1...m, 1...p]) {
   Sor i=7 to n do
        for 1=1 to P do
           C[1,7]=0;
                 C[i, i] - c[i, i] + A[i, k] * 13[k,j];
```

ضرب دو ماتریس غیر مربعی

Matrix Multiply (n, m, p, A[1...n, 1...m], 13[1...m, 1...p]) { for i=7 to n do for 1=1 to P do C[1,7]=0; C[i,j] = c[i,j] + A[i,k] * 13[k,j]; - / (n, m, p) = n x m x p مرتب سازی تعویضی (Exchange Sort)

مرتب سازی تعویضی (Exchange Sort)

Exchange-Sort (n, A[1..n]) $\{$ $\{$ $\{$ $\}$

Exchange-Sort (n, A[1..n]) مرتب سازی تعویضی 800 i=1 to n-1 do for 7=i+1 to n do if ACiT > ACTIT Swap (A[i], A[i]);

• عمل اصلى: ؟

Exchange-sort (n, A[1..n]) $\{$ $\{$ $\{$ $\{$ $\}$ $\{$ $\{$ $\}$ $\{$ $\}$ $\{$ $\{$ $\}$ $\{$ $\}$ $\{$ $\{$ $\}$ $\{$ $\}$ $\{$ $\{$ $\}$ $\{$ $\{$ $\}$ $\{$ $\{$ $\}$ $\{$ $\}$ $\{$ $\{$ $\}$ $\{$ $\{$ $\}$ $\{$ $\{$ $\}$ $\{$ $\{$ $\}$ $\{$ $\{$ $\}$ $\{$ $\{$ $\}$ $\{$ $\{$ $\}$ $\{$ $\{$ $\}$ $\{$ $\{$ $\}$ $\{$ $\{$ $\}$ $\{$ $\{$ $\}$ $\{$ $\{$ $\}$ $\{$ $\{$ $\}$ $\{$ $\{$ $\}$ $\{$ $\{$ $\}$ $\{$ $\{$ $\}$ $\{$ $\{$ $\}$ $\{$ $\{$ $\}$ $\{$

• عمل اصلى: مقايسه عناصر

Exchange-Sort
$$(n, AE1..n7)$$
?

Sor $i=1$ to $n-1$ do

For $j=i+1$ to n do

if $AEi7 > AEj7$

Swap $(AEi], AEj7)$;

• عمل اصلى: مقايسه عناصر

$$(n-1)(n-(i+1)+1)=(n-1)(n-i)$$
 !!!!!!

Exchange-Sort
$$(n, AE1..n7)$$
?

Sor $i=1$ to $n-1$ do

For $j=i+1$ to n do

if $AEi7 > AEj7$

Swap $(AEi], AEj7)$;

• عمل اصلى: مقايسه عناصر

$$(n-1)(n-(i+1)+1)=(n-1)(n-i)$$
 !!!!!!

Exchange-Sort (n, A[1..n]) 800 i=1 to n-1 do for 3=i+1 to n do if ACi]> ACi] Swap (A[i], A[i]);

مرتب سازی تعویضی

i institute

Exchange-Sort (n, A[1..n]) 800 i=1 to n-1 do for 3=i+1 to n do if ACi]>ACi] Swap (A[i], A[i]);

مرتب سازی تعویضی

i 3 west sta

Exchange-Sort (n, A[1..n]) 800 i=1 to n-1 do for 3=i+1 to n do if ACi]>ACi] Swap (A[i], A[i]);

مرتب سازی تعویضی

2 n-1 2 n-2 Exchange-Sort (n, A[1..n]) 800 i=1 to n-1 do for 7=i+1 to n do if ACi]>ACi] Swap (A[i], A[i]);

i	3 west state
1	n-1
2	n-2
	11-0

Exchange-Sort (n, A[1..n]) 800 i=1 to n-1 do for 7=i+1 to n do if ACi]>ACi] Swap (A[i], A[i]);

i	3 west sta
1	n-1
2	n-2
3	n-3
:	
n-7	1

Exchange-Sort (n, A[1..n]) 800 i=1 to n-1 do for 7=i+1 to n do if ACi]>ACi] Swap (A[i], A[i]);

i	3 west fales
1	n-1
2	n-2
3	n-3
:	
n-7	1
-	·

Exchange-Sort
$$(n, A[1..n])$$
 {

8or $i=1$ to $n-1$ do

8or $j=i+1$ to n do

if $A[i] > A[j]$

5wap $(A[i], A[j])$;

i	3 west state
1	n-1
2	n-2
3	n-3
1	
n-7	7

Exchange-Sort
$$(n, A[1..n])$$
 {

8or $i=1$ to $n-1$ do

8or $j=i+1$ to n do

if $A[i] > A[j]$

5wap $(A[i], A[j])$;

کته:

```
Sor (= 1 to n do
```

نکته:

Sor
$$i=1$$
 to n do $\begin{cases} \sum_{i=1}^{n} k(i) \end{cases}$

Exchange-Sort (n, A[1..n]) مرتب سازی تعویضی 800 i=7 to n-1 do for 3=i+1 to n do if ACi]>ACi] Swap (A[i], A[i]);)

Exchange-Sort (n, A[1..n]) مرتب سازی تعویضی 800 i=1 to n-1 do for 7=i+1 to n do if ACi3> ACi3? > n-c Swap (A[i], A[i]);)

=>
$$7(n) = \sum_{i=1}^{n-1} (n-i)$$

Exchange-Sort (n, A[1..n]) مرتب سازی تعویضی 808 i= 7 to n-1 do for 7=i+1 to n do if ACi3> ACi3] > n-c Swap (A[i], A[i]);)

=>
$$7(n) = \sum_{i=1}^{n-1} (n-i) = \frac{n(n-1)}{2}$$