طراحى الگوريتم ها پيچيدگي الگوريتم ها(ادامه)

استاد درس: مهدی جبل عاملی

```
index linear-search (n, A[1.on],x)}
     C=1;
   while (i kn) and (A [i] = x)
    if (c/n)
     return i; 4
```

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

• عمل اصلى: مقايسه عناصر

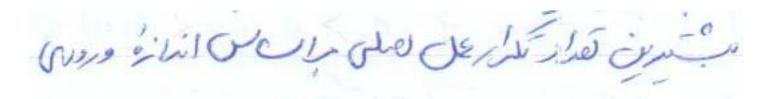
```
index linear-search (n, A[1.on],x)}
     C=1;
   while (i kn) and (A [i] = x)
     if (c>n)
     return i; 3
```

• عمل اصلى: مقايسه عناصر

T(n) = ?????

زمان اجرا در بدترین حالت (Worst-case)

W(n)



```
index linear-search (n, A[1.on],x)}
     L=1;
   while (i kn) and (A [i] ≠ x)
    if (ixn)
     return i; 3
                               W(n)=
                                    55555
```

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

W(n) = n

الگوریتم جستجوی دودویی (مر [۱۰۰۱] Binary-Search (مر الگوریتم جستجوی دودویی) 1=1; H=n; while (L < H) M = |(L + H)/2|;if K=A[M] return M; if x < A[M] H=M-1; 2 =M+(; return o;

• عمل اصلى: مقايسه عناصر

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 K=A[M] return M; if x < A[N] H=M-1; 2 = M + (; else return 0; }

• عمل اصلى: مقايسه عناصر

$$T(n) = ?????$$

الگوریتم جستجوی دودویی (مر [۱۰۰۱] الگوریتم جستجوی دودویی ا 1=1; H=n; while (L < H) M= ((++)/2); if K=A[M] return M; if x < A[M] H=M-1; 2 = M + 1 ; else return 0; }

• عمل اصلى: مقايسه عناصر

W(n)=**55555**

الكوريتم جستجوى دودويي (مر [۱۰۰۱] Binary-Search (مر الكوريتم جستجوى دودويي) 1=1; H=n; while (L < H) $M = \lfloor (L + H)/2 \rfloor;$ if n=A[M] return M; if x < A[M] H=M-1; 2 =M+1; return 0; }

• عمل اصلى: مقايسه عناصر

ارتباط بين T(n) و W(n)

ارتباط بين T(n) و W(n)

• اگر T(n)=W(n) وجود داشته باشد

ارتباط بین T(n) و W(n)

- اگر T(n)=W(n) وجود داشته باشد
 - در غیر این صورت، !!!!

مقایسه جست و جوی خطی و دودویی در بدترین حالت

• جست و جوى خطى: n

$$W(n) = \left(\left\lfloor \log_2 n \right\rfloor + 1 \right) \times 2$$
 : $(n) = \left(\left\lfloor \log_2 n \right\rfloor + 1 \right)$

مقایسه جست و جوی خطی و دودویی در بدترین حالت

• جست و جوى خطى: n

$$W(n) = \left(\left\lfloor \log_2 n \right\rfloor + 1 \right) \times 2$$
 : $(n) = (\log_2 n) \times 1$

زمان اجرا در بهترین حالت (Best-case)

B(n) Guy dilloch del de la les com

زمان اجرای الگوریتم جستجوی خطی در بهترین حالت

```
index linear-search (n, A[1.on],x)}
     C=1;
   while (i kn) and (A [i] = x)
    if (in)
     return i; 4
                               B(n)=
                                   55555
```

زمان اجرای الگوریتم جستجوی خطی در بهترین حالت

```
index linear-search (n, A[1.on],x)}
     C=1;
   while (i kn) and (A [i] = x)
    if (c>n)
     return i; 4
```

$$B(n)=1$$

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+(; else return 0; }

$$B(n) = ?????$$

الگوریتم جستجوی دودویی (مر [۱۰۰۱] Binary-Search (مر الگوریتم جستجوی دودویی) 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; else return 0; }

$$B(n)=1$$

مقایسه جست و جوی خطی و دودویی

$$B(n)=1$$
 $W(n)=n$ عصلی: $w(n)=n$

زمان اجرا در حالت متوسط (Average-case)

A(n) ouis delete del de la desir de cur lu

```
index linear-search (n, A[1...n], x)}
   while (i kn) and (A [i] = x)
    return i; 4
```

```
index linear-search (n, A[1...n], x)}
   while (i kn) and (A [i] = x)
    return i; 4
```

• الف – فرض كنيد كليد در بين عناصر وجود دارد.

index linear-search (n, A[1...n], x)} while (i kn) and (A [i] ≠ x) return i; }

• الف- فرض كنيد كليد در بين عناصر وجود دارد.

به احتمال $\frac{1}{n}$ کلید در خانه i ام است.

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

• الف- فرض كنيد كليد در بين عناصر وجود دارد.

به احتمال $\frac{1}{n}$ کلید در خانه \mathbf{i} ام است.

• الف- فرض كنيد كليد دربين عناصر وجود دارد.

به احتمال $\frac{1}{n}$ کلید در خانه i ام است.

index linear-search
$$(n, A[1..n], x)$$
?

 $i=1;$

while $(i \le n)$ and $(A[i] \ne x)$
 $i+1;$

if $(i \ge n)$
 $i=p;$

return i ; $i \ge a$
 $i \ge$

• الف- فرض كنيد كليد در بين عناصر وجود دارد.

به احتمال $\frac{1}{n}$ کلید در خانه \mathbf{i} ام است.

index linear-search
$$(n, A[1..n], x)$$
?

 $i=1;$

while $(i \le n)$ and $(A[i] \ne x)$

if $(i \ge n)$
 $i=0;$

return i ; i ?

 $i = 0$;

 $i = 0$;

• الف- فرض كنيد كليد در بين عناصر وجود دارد.

به احتمال $\frac{1}{n}$ کلید در خانه \mathbf{i} ام است.

index linear-search
$$(n, A[1...n], x)$$
 is index linear-search $(n, A[1...n], x)$ in the city and $(A[i] \neq x)$ if $(i \nmid n)$ if $(i \mid n)$

$$A(n) = \frac{n}{\sum_{i=1}^{n} \frac{1}{n} \times i} = \frac{1}{n} \left(\frac{n(n+1)}{2} \right)$$

• الف - فرض كنيد كليد در بين عناصر وجود دارد. إلى أيام index linear-search (n, A[1...] الف - فرض كنيد كليد در بين عناصر وجود دارد.

به احتمال $\frac{1}{n}$ کلید در خانه \mathbf{i} ام است.

به میزان i مقایسه لازم است تا کلید پیدا شود.

inear-search
$$(n, H[1..n], x)$$

 $i=1;$
while $(i \le n)$ and $(A[i] \ne x)$
 $i+1;$
 $i = 0;$

return i; }

$$A(n) = \frac{n}{2} \frac{1}{n} \times i = \frac{1}{n} \left(\frac{n(n+1)}{2} \right)$$

 $i = 1$

$$\Rightarrow A(n) = \frac{n+1}{2}$$

while (i kn) and (A [i] = x)

return i; }

• ب- فرض کنید کلید به احتمال p در بین عناصر و جود دارد. $\{x, y\}$ میال p استمال p در بین عناصر و جود دارد.

• ب- فرض کنید کلید به احتمال p در بین عناصر و جود دارد. عناصر و جود دارد. از ۱،۰۰۱ مین عناصر و استان p استان م

while (i kn) and (A [i] = x)

return i; &

• ب- فرض کنید کلید به احتمال p در بین عناصر و جود دارد. عناصر و اسم الم الم الم الم الم index linear-search الم

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

ار تباط بین T(n) و سه زمان اجرای دیگر

- اگر T(n)=W(n)=B(n)=A(n) وجود داشته باشد
 - در غير اين صورت، !!!!