

# فشرده‌سازی اطلاعات

۰۱-۷۰۲-۱۰-۱۴۰

بخش دوم



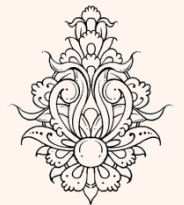
دانشگاه شهید بهشتی  
پژوهشکده‌ی فضای مجازی

بهار ۱۳۹۷

احمد محمودی ازناوه

# فهرست مطالب

- معرفی
- روش‌های ایستا
  - کدهای دوتایی
- روش‌های وفقی (پویا)
  - الگوریتم LZ77
  - الگوریتم LZ78
  - الگوریتم LZW
- چند نمونه از کاربردهای کدگذاری واژه‌نامه‌ای



# معرفی

- در روش‌هایی که تاکنون بررسی شد، فرض بر این بود که نمادها به صورت مستقل از یک توزیع تصادفی پیروی می‌کنند (i.i.d.).
  - اغلب بین نمادهای مختلف نوعی **همبستگی** وجود دارد.
  - معمولاً پیش از کدگذاری یک مرحله **ناهمبسته‌سازی** (decorrelation) انجام می‌شود.
- در این بخش به مرور روشی می‌پردازیم که **ساختار داده‌ها** را نیز در نظر می‌گیرد.
- برای دنباله‌ای از نمادها که زیاد تکرار می‌شوند، یک کد در نظر گرفته می‌شوند.
  - چنین دنباله‌هایی در متون و دستوره‌های کامپیوتری زیاد یافت می‌شود.
  - در زندگی روزمره هم با چنین رویکردی مواجه می‌شویم!



# روش‌های مبتنی بر واژه‌نامه

- در این شیوه واژه‌نامه‌ای از دنباله‌های پرتکرار تهیه شده و به جای کد کردن تک‌تک نمادهای چنین دنباله‌هایی، شاخص آن در واژه‌نامه در نظر گرفته می‌شود.
  - سایر نمادها به صورت تکی کدگذاری خواهند شد.
- برای این که این شیوه کارایی لازم را داشته باشد، تعداد مجموعه نمادهای پرتکرار و در نتیجه اندازهی واژه‌نامه باید به صورت نسبی کوچک باشد.
- متنی شامل کلمه‌های چهارحرفی از متشکل از ۲۶ حرف کوچک انگلیسی و ۶ نماد برای علامت‌گذاری
  - ۲۵۶ کلمه پرتکرار در واژه‌نامه‌ی گردآوری می‌شوند.
  - در صورت استفاده از واژه‌نامه کلماتی که از شاخص واژه‌نامه استفاده می‌کنند با پیشوند «1» و سایر کلمه‌کدها با «0» آغاز می‌شوند.

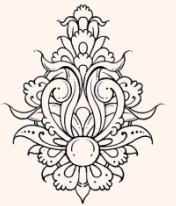


$$R = 9p + (1 - p)21 = 21 - 12p \leq 20$$

the average number of bits per pattern

# روش‌های ایستا

- در این دسته از روش‌های یک واژه‌نامه‌ی **ثابت** در نظر گرفته می‌شود.
- استفاده از چنین روش‌های زمانی مفید است که از خصوصیات نمادها اطلاعات کاملی داشته باشیم.  
– این ویژگی‌ها به مرور زمان تغییر نکند.
- چنین روش‌هایی معمولاً در کاربردهای خاصی می‌تواند مفید باشد.  
– به عنوان مثال در فشرده‌سازی مستندات مربوط به اطلاعات دانشجویان واژه‌هایی مانند شماره دانشجویی، معدل، واحدهای پاس‌شده، مشروطی و ... زیاد تکرار می‌شوند.



- در این شیوهی دنباله‌های دوتایی پرتکرار کد می‌شوند.  
– به عنوان نمونه برای کدگذاری کاراکترهای قابل چاپ اسکی، می‌توان یک جدول ۲۵۶ تایی در نظر گرفت، ۹۵ مدخل آن به کاراکترهای قابل چاپ اختصاص یابد و مابقی آن به دنباله‌های دوتایی پرتکرار

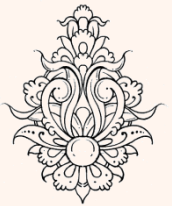
$$A = \{a, b, c, d, r\}$$

Code	Entry	Code	Entry
000	<i>a</i>	100	<i>r</i>
001	<i>b</i>	101	<i>ab</i>
010	<i>c</i>	110	<i>ac</i>
011	<i>d</i>	111	<i>ad</i>

*abracadabra*

101 – 100 – 110 – 111 – 101 – 100 – 000

مثال



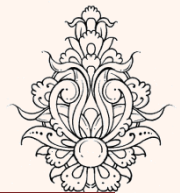
## LaTeX document

Pair	Count	Pair	Count
<i>el</i>	1128	<i>ar</i>	314
<i>lt</i>	838	<i>at</i>	313
<i>ll</i>	823	<i>lw</i>	309
<i>th</i>	817	<i>te</i>	296
<i>he</i>	712	<i>ls</i>	295
<i>in</i>	512	<i>dl</i>	272
<i>sl</i>	494	<i>lo</i>	266
<i>er</i>	433	<i>io</i>	257
<i>la</i>	425	<i>co</i>	256
<i>tl</i>	401	<i>re</i>	247
<i>en</i>	392	<i>lS</i>	246
<i>on</i>	385	<i>rl</i>	239
<i>nl</i>	353	<i>di</i>	230
<i>ti</i>	322	<i>ic</i>	229
<i>li</i>	317	<i>ct</i>	226

## collection of C programs

Pair	Count	Pair	Count
<i>ll</i>	5728	<i>st</i>	442
<i>nl</i>	1471	<i>le</i>	440
<i>;nl</i>	1133	<i>ut</i>	440
<i>in</i>	985	<i>f(</i>	416
<i>nt</i>	739	<i>ar</i>	381
<i>= l</i>	687	<i>or</i>	374
<i>li</i>	662	<i>rl</i>	373
<i>tl</i>	615	<i>en</i>	371
<i>l =</i>	612	<i>er</i>	358
<i>);</i>	558	<i>ri</i>	357
<i>, l</i>	554	<i>at</i>	352
<i>nl</i>	506	<i>pr</i>	351
<i>lf</i>	505	<i>te</i>	349
<i>el</i>	500	<i>an</i>	348
<i>l*</i>	444	<i>lo</i>	347

مطلوب است روشی اتخاذ شود که واژه‌نامه بر اساس  
ممتوای نمادها به صورت وفقی تشکیل شود.





Jacob Ziv



# روش‌های وفقی مبتنی بر واژه‌نامه

• واژه‌نامه، بخشی از داده‌های کد شده است.

## The LZ77 Approach

Jacob Ziv and Abraham Lempel

Sliding window

Match pointer

Length of match



لازم است کدگذاری شوند

Search buffer

offset

Look ahead buffer

$\langle o, l, c \rangle$

کدگذاری شده

$o$ : offset

$l$ : length of match

$c$ : codeword of the symbol in the look-ahead buffer following the match

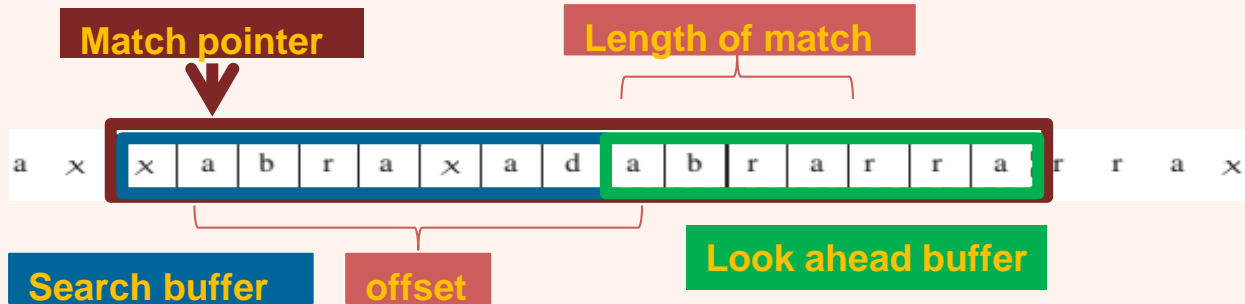
شیوهی کدگذاری





# تمرین

- سه تایی به دست آمده برای نمایش با طول ثابت  
به چند بیت نیاز دارد؟  
 $\langle o, l, c \rangle$

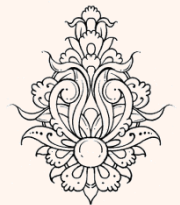


$$\lceil \log_2^S \rceil + \lceil \log_2^W \rceil + \lceil \log_2^A \rceil$$

اندازه‌ی پنجره‌ی جستجو

اندازه‌ی پنجره

تعداد مروف الفبا



طول پنجره: ۱۳ طول بافر پیش‌بینی: ۶

رشته‌ی زیر را در نظر بگیرید:

... *cabracadabrarrarrad* ...

*cabraca* *dabrarr*

$\langle 0, 0, C(d) \rangle$

*abracad* *abrarr*

$\langle 7, 4, C(r) \rangle$

*adabrarr* *rarrad*

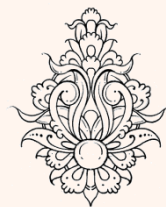
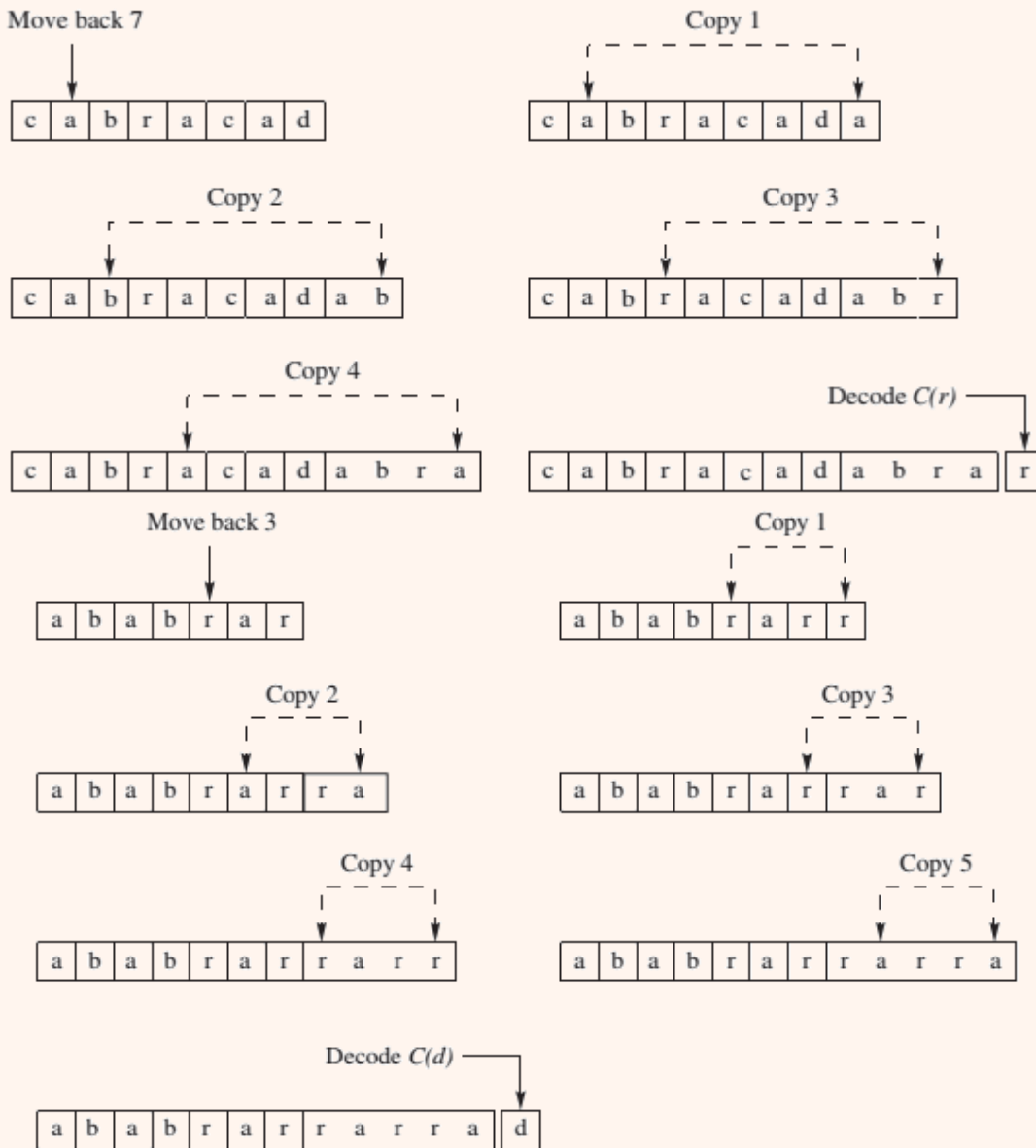
$\langle 3, 5, C(d) \rangle$

کد به دست آمده بر اساس LZ77

*cabraca*  $\langle 0, 0, C(d) \rangle \langle 7, 4, C(r) \rangle \langle 3, 5, C(d) \rangle$

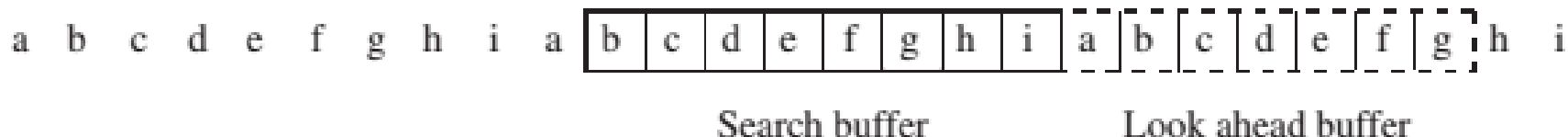


cabraca  $\langle 0,0,C(d) \rangle \langle 7,4,C(r) \rangle \langle 3,5,C(d) \rangle$



- چنانچه تکرار داده‌ها با تناوبی بیشتر از طول پنجره رخ دهد، LZ77 نه تنها کارایی خود را از دست می‌دهد، بلکه به جای فشرده‌سازی بر حجم داده‌ی گذشته می‌افزاید.

مثال

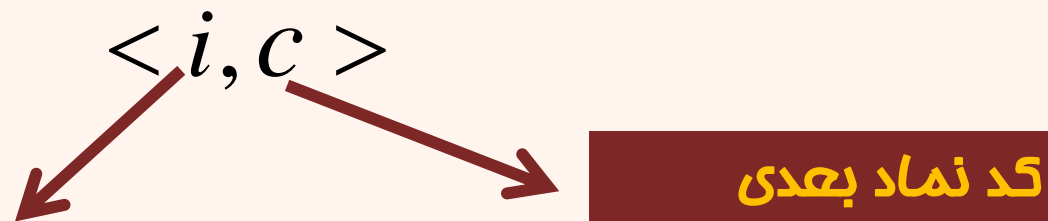


- ویرایش‌های بهبود یافته‌ی متعددی از LZ77 مطرح شده است.

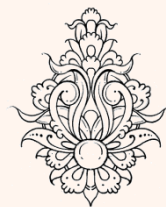
- بیشتر بر مبنای شیوه‌ی کدگذاری سه‌تایی‌ها هستند.
- بسیاری از فشرده‌سازها نظیر ZIP از این شیوه‌ها استفاده می‌کنند.



- LZ78 نیز راهی برای پیروی بر مشکلات LZ77 است. در این شیوه واژه‌نامه به صورت مشابه در کدگذار و کدگشا ساخته می‌شود.
- در این شیوه داده‌ها به صورت زیر کد می‌شوند:



- در صورتی نیافتن هیچ مشابهی در واژه‌نامه از شافص «0» استفاده می‌شود.



*wabba wabba wabba wabba woo woo woo*

در ابتدا واژه‌نامه خالی است و چند نماد ابتدایی با شاخص 0 مشخص می‌شوند.

$\langle 0, C(w) \rangle$

$\langle 0, C(a) \rangle$

$\langle 0, C(b) \rangle$

واژه‌نامه

Index	Entry
1	<i>w</i>
2	<i>a</i>
3	<i>b</i>

چهارمین نماد «*b*» در واژه‌نامه وجود دارد، نماد بعدی را به آن الماق کنیم:  
«*ba*» و به صورت زیر کد می‌شود:

$\langle 3, C(a) \rangle$



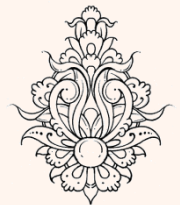
# ادامه‌ی مثال

و به همین ترتیب دنباله‌نمادهای دیگر کد شده و هر چه پیش روی نمادهای طولانی‌تری انتخاب می‌شوند. در پایان خواهیم داشت:

Encoder Output	Index	Dictionary Entry
$\langle 0, C(w) \rangle$	01	$w$
$\langle 0, C(a) \rangle$	02	$a$
$\langle 0, C(b) \rangle$	03	$b$
$\langle 3, C(a) \rangle$	04	$ba$
$\langle 0, C(\phi) \rangle$	05	$\phi$
$\langle 1, C(a) \rangle$	06	$wa$
$\langle 3, C(b) \rangle$	07	$bb$
$\langle 2, C(\phi) \rangle$	08	$a\phi$
$\langle 6, C(b) \rangle$	09	$wab$
$\langle 4, C(\phi) \rangle$	10	$ba\phi$
$\langle 9, C(b) \rangle$	11	$wabb$
$\langle 8, C(w) \rangle$	12	$a\phi w$
$\langle 0, C(o) \rangle$	13	$o$
$\langle 13, C(\phi) \rangle$	14	$o\phi$
$\langle 1, C(o) \rangle$	15	$wo$
$\langle 14, C(w) \rangle$	16	$o\phi w$
$\langle 13, C(o) \rangle$	17	$oo$



*wabba wabba wabba wabba woo woo woo*



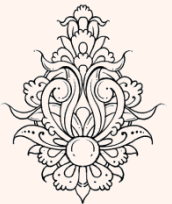
a modification by Terry Welch

• در این شیوه تنها بخش شاخص ارسال می‌شود.

< i  >

• واژه‌نامه‌ی ابتدایی باید شامل تمام الفبا باشد.

• در صورت وجود ترکیب‌های طولانی در دنباله‌ی داده‌ها این ترکیبات به واژه‌نامه افزوده می‌شود.





# مثال-کدگذاری

*wabba wabba wabba wabba woo woo woo*

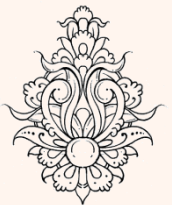
الفبای کد منبع شامل حروف زیر است:

$\{ \emptyset, a, b, o, w \}$

در نتیجه واژه‌نامه‌ی اولیه به صورت زیر خواهد بود:

Index	Entry
1	$\emptyset$
2	<i>a</i>
3	<i>b</i>
4	<i>o</i>
5	<i>w</i>

در کدگذار ابتدا با «*w*» مواجه می‌شود که در واژه‌نامه موجود است، از این رو «*wa*» انتخاب می‌شود و به عنوان ششمین مدخل به واژه‌نامه افزوده می‌شود.

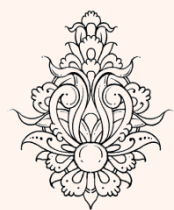


سپس از حرف «a» ادامه می‌دهیم و «ab» را به واژه‌نامه می‌افزاییم.

همین روند را ادامه می‌دهیم تا به اولین حرف از دومین کلمه برسیم، تا اینجا رشته‌ی کد شده عبارتست از :

5 2 3 3 2 1

Index	Entry
01	∅
02	a
03	b
04	o
05	w
06	wa
07	ab
08	bb
09	ba
10	a∅
11	∅w
12	w...



wabba wabba wabba wabba woo woo woo

# مثال - کدگذاری

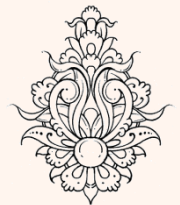
در کدگذار ابتدا با «w» مواجه می‌شود که در واژه‌نامه موجود است، از این رو «wa» انتخاب می‌شود، که آن هم در کدگذار موجود است. در نتیجه «wab» انتخاب می‌شود و به واژه‌نامه افزوده می‌شود.

*wabba wabba wabba wabba woo woo woo*

کدگذار 6 را در ادامه‌ی دنباله ارسال خواهد کرد.

در نهایت واژه‌نامه و رشته‌ی کد شده به صورت زیر خواهند بود:

Index	Entry	Index	Entry
01	∅	14	a∅ w
02	a	15	wabb
03	b	16	ba∅
04	o	17	∅ wa
05	w	18	abb
06	wa	19	ba∅ w
07	ab	20	wo
08	bb	21	oo
09	ba	22	o∅
10	a∅	23	∅ wo
11	∅ w	24	oo∅
12	wab	25	∅ woo
13	bba		



5 2 3 3 2 1 6 8 10 12 9 11 7 16 5 4 4 11 21 23 4

# مثال- کدگشایی

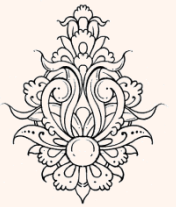
5 2 3 3 2 1 6 8 10 12 9 11 7 16 5 4 4 11 21 23 4

Index	Entry
1	$\emptyset$
2	<i>a</i>
3	<i>b</i>
4	<i>o</i>
5	<i>w</i>

همزمان با کدگشایی واژه نامه نیز تشکیل می شود:

wabba 6 8 10 12 9 11 7 16 5 4 4 11 21 23 4

Index	Entry
01	$\emptyset$
02	<i>a</i>
03	<i>b</i>
04	<i>o</i>
05	<i>w</i>
06	<i>wa</i>
07	<i>ab</i>
08	<i>bb</i>
09	<i>ba</i>
10	<i>a<math>\emptyset</math></i>
11	$\emptyset$



wabba wabba wabba wabba woo woo woo

# مثال - کدگذاری

*ababababababababababababab...*

الفبای کد منبع شامل حروف زیر است:

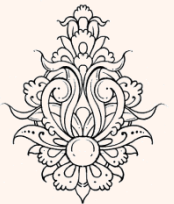
$\{a, b\}$

در نتیجه واژه‌نامه‌ی اولیه به صورت زیر خواهد بود:

Index	Entry
1	<i>a</i>
2	<i>b</i>

1 2 3 5 ...

Index	Entry
1	<i>a</i>
2	<i>b</i>
3	<i>ab</i>
4	<i>ba</i>
5	<i>aba</i>
6	<i>abab</i>
7	<i>b...</i>



# مثال - کدگذاری

1 2 3 5 ...

Index	Entry
1	<i>a</i>
2	<i>b</i>

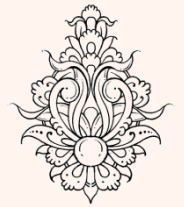
*ab* 3 5 ...

Index	Entry
1	<i>a</i>
2	<i>b</i>
3	<i>ab</i>
4	<i>b...</i>

*abab* 5 ...

Index	Entry
1	<i>a</i>
2	<i>b</i>
3	<i>ab</i>
4	<i>ba</i>
5	<i>a...</i>

واژه نامه ناقص



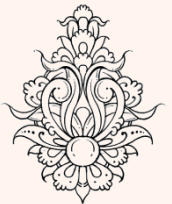
LZW نیاز به یک سیستم مدیریت استثنای هم دارد!

- از الگوریتم‌های مبتنی بر LZ، در عمل فراوان مورد استفاده قرار گرفته‌اند. به عنوان مثال

– دستور compress در سیستم عامل UNIX

LZW

- اندازه‌ی واژه‌نامه، وفقی است و از ۵۱۲ شروع می‌شود. (بیت برای شاخص).
- با پر شدن واژه‌نامه، اندازه‌ی آن دو برابر می‌شود.
- وقتی به واژه‌نامه حداکثر اندازه رسید(شانزده بیت برای شاخص)، عملاً به یک روش ایستا تبدیل می‌شود.
- در این حالت نرخ فشردگی بررسی شده و چنانچه از یک حد آستانه کمتر بود، واژه‌نامه تخلیه شده و از نو ساخته می‌شود.



256×256

# تصاویر نمونه



Earth



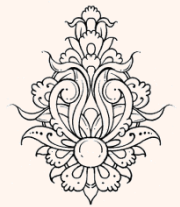
Omaha



Sensin



Sena





# کاربردها (ادامه...)

## Graphics Interchange Format

### – فرمت فشرده‌سازی GIF

**LZW**

- شبیه به compress است.
- برای تصاویر طبیعی چندان مناسب نیست.

Image	GIF	Arithmetic Coding of Pixel Values	Arithmetic Coding of Pixel Differences
Sena	51,085	53,431	31,847
Sensin	60,649	58,306	37,126
Earth	34,276	38,248	32,137
Omaha	61,580	56,061	51,393

## Portable Network Graphics

### – فرمت فشرده‌سازی PNG

- با توجه به حق امتیاز LZW به عنوان جایگزین GIF مطرح شد.

**LZW**

### – فرمت فشرده‌سازی V.42 bis در ارتباطات تلفنی

