بینایی کامپیوتر
فصل سوم
فشرده‌سازی ۱
COMPRESSION

محمدجواد فدائی اسلام

بهمن ۱۴۰۳

- فشرده سازی:

کاهش داده لازم برای نمایش اطلاعات یک تصویر، فشرده سازی نام دارد.

اطلاعات با داده متفاوت است.

داستانی که توسط انسان پرحرف گفته می شود داده های آن زیاد است. اما اطلاعات آن ثابت است.

نسبت فشرده سازی:

$$C=\frac{n_1}{n_2}$$

n_1=نمایش اولیه اطلاعات

n_2=نمایش ثانویه اطلاعات

افزونگی:

$$R = 1 - \frac{1}{C}$$

چه مقدار داده برای نمایش اطلاعات یا تصویر مورد نیاز است؟

- تعداد تکرار نماد a_j درتصویر $P(a_j)=$
- اطلاعات موجود در این نماد $I(a_j) = -\log P(a_j)$
- متوسط اطلاعات درتصویر یا آنتروپی

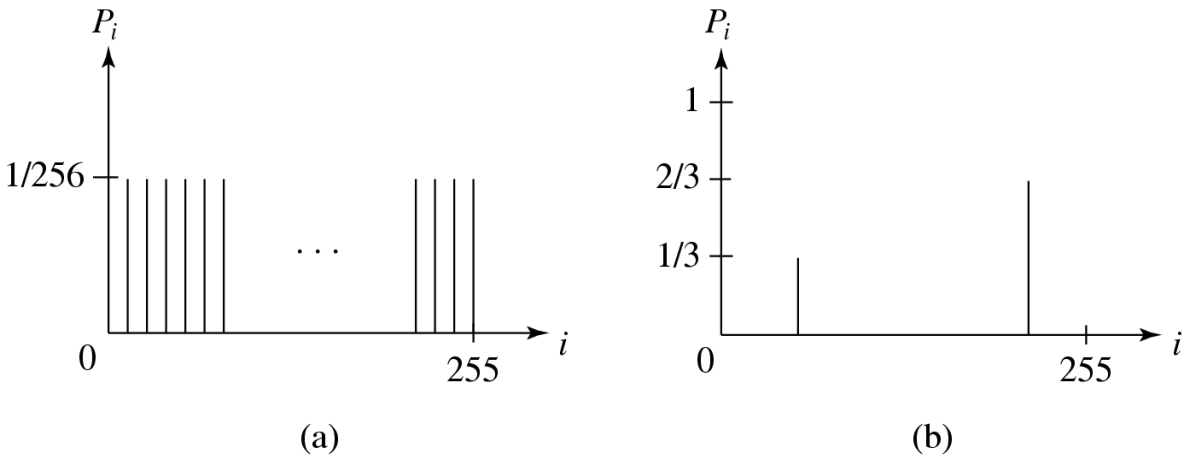
$$\eta = -\sum_{j=0}^{J} P(a_j) \log P(a_j)$$

آنتروپی

- آنتروپی میزانی برای مقدار اطلاعات موجود در یک منبع است.
- آنتروپی η حد پایین برای میانگین تعداد بیت‌ها برای کدکردن هر مجموعه سمبل S را مشخص می‌کند (با فرض آنکه علایم ترکیب نشوند).

$$\eta \leq \bar{l}$$

- نماد $\bar{l}$ نشان دهنده طول میانگین (بر حسب بیت) کدهای تولید شده توسط یک کدکننده است.



- (a) هیستوگرام یک تصویر با شدت‌های توزیع سطح خاکستری یک‌نواخت و مساوی را نشان می‌دهد، به ازای هر i داریم:

$p_i = 1/256$ بنابراین آنتروپی این تصویر برابر است با:

$$\log_2 256 = 8$$

$$\sum_{i=1}^{256} \frac{1}{256} \log_2 256 = 8$$

- (b) هیستوگرام یک تصویر با دو مقدار ممکن را نشان می‌دهد. آنتروپی آن ۰,۹۲ است.

$$\frac{1}{3}\log_2 3 + \frac{2}{3}\log_2 \frac{3}{2}$$

انواع افزونگی

- مبتنی بر کد(coding redundancy)
- مبتنی بر اطلاعات فضایی یا بین پیکسلی (Interpixel redundancy)
- دادههایی که چشم به آنها حساس نیست (Psychovisual redundancy)

نمایش سه تصویر با سه نوع افزونگی

افزونگی کد | افزونگی بین پیکسلی | افزونگی به علت عدم حساست چشم

a b c

FIGURE 8.1 Computer generated 256 × 256 × 8 bit images with (a) coding redundancy, (b) spatial redundancy, and (c) irrelevant information. (Each was designed to demonstrate one principal redundancy but may exhibit others as well.)

تعداد بیت لازم برای کدگذاری طول متغیر

Coding Redundancy

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_{87} = 87$	0.25	01010111	8	01	2
$r_{128} = 128$	0.47	10000000	8	1	1
$r_{186} = 186$	0.25	11000100	8	000	3
$r_{255} = 255$	0.03	11111111	8	001	3
r_k for $k \neq 87, 128, 186, 255$	0	—	8	—	0

$$L_{avg} = 0.25(2) + 0.47(1) + 0.25(3) + 0.03(3) = 1.81 \text{ bits}$$

$$C = \frac{256 \times 256 \times 8}{118{,}621} = \frac{8}{1.81} \approx 4.42$$

$$R = 1 - \frac{1}{4.42} = 0.774$$

تعداد بیت لازم برای کدگذاری طول ثابت برابر ۸ است.

Spatial and Temporal Redundancy

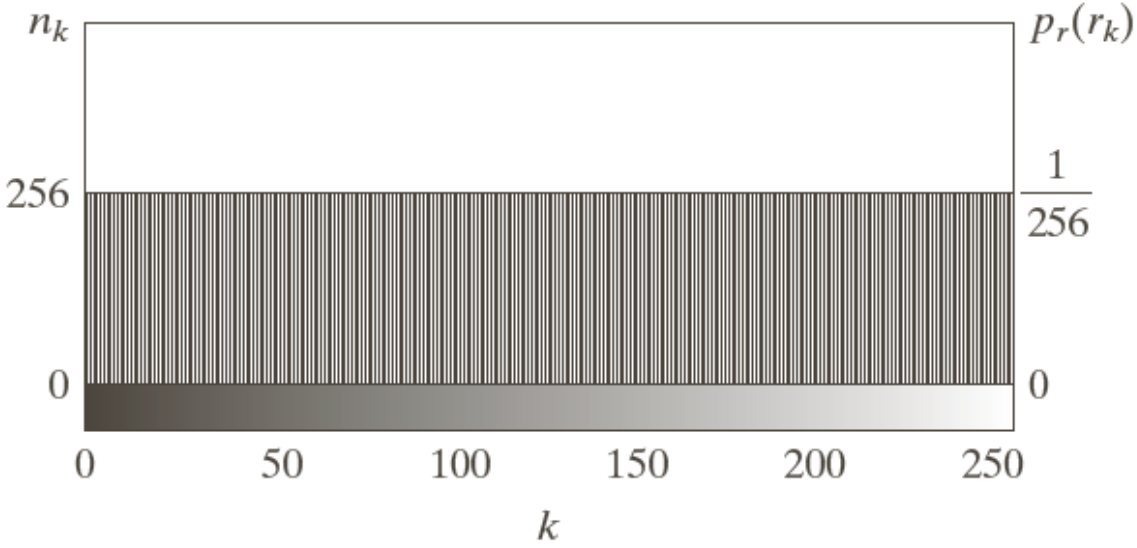
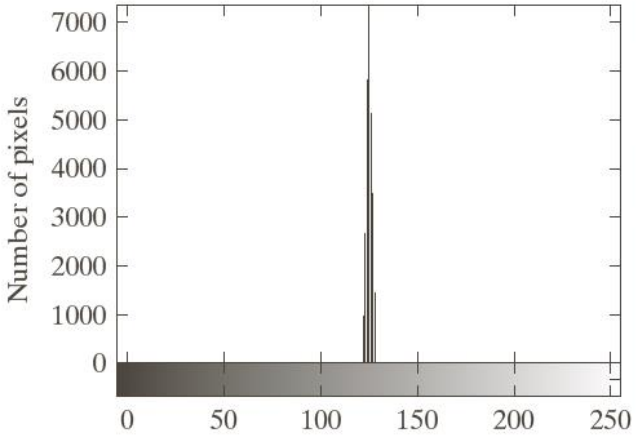
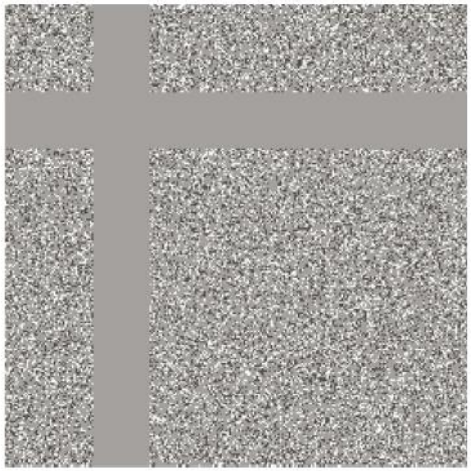


FIGURE 8.2 The intensity histogram of the image in Fig. 8.1(b).

Run-length coding

تصویر اصلی، افزونگی به علت
عدم حساست چشم





a b

FIGURE 8.3
(a) Histogram of the image in Fig. 8.1(c) and (b) a histogram equalized version of the image.

داده های متنوعی که بر اثر نرمال سازی هیستوگرام قابل رویت شده اند.
اما در تصویر اصلی قابل رویت نیستند.

مراحل فشرده سازی

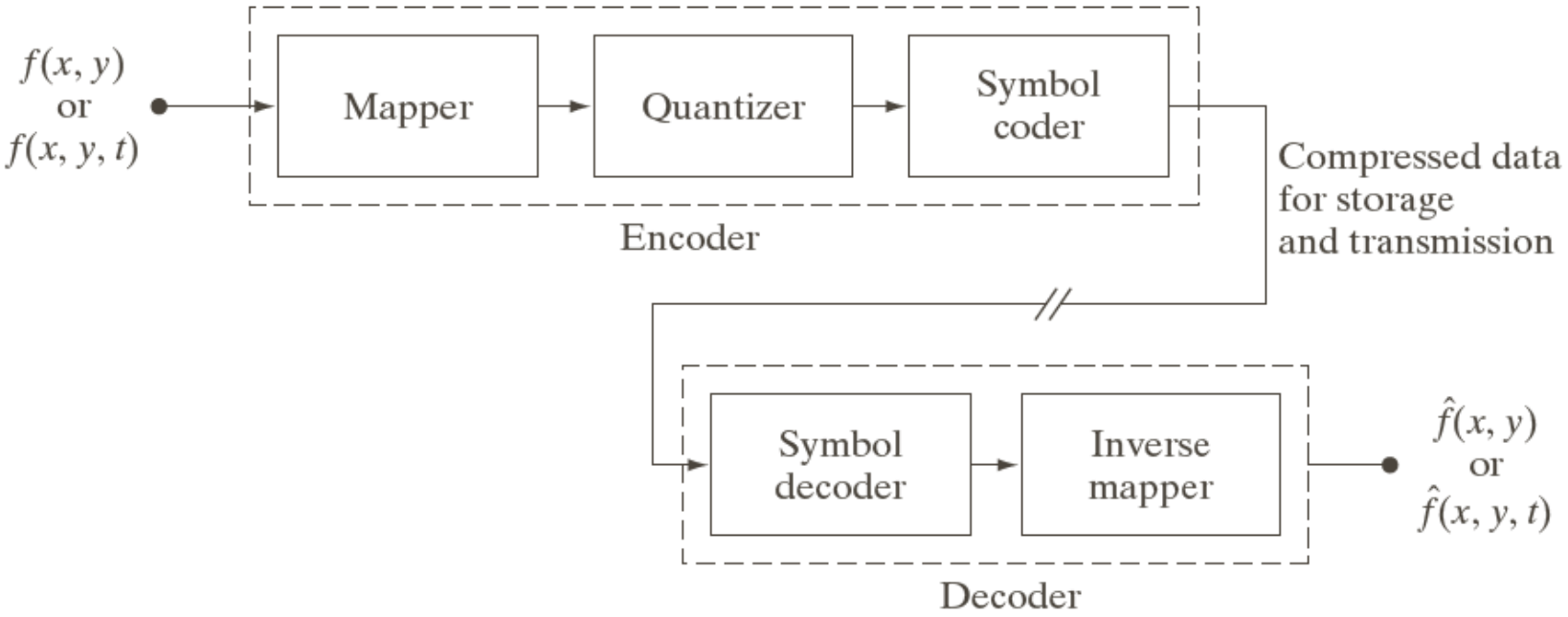


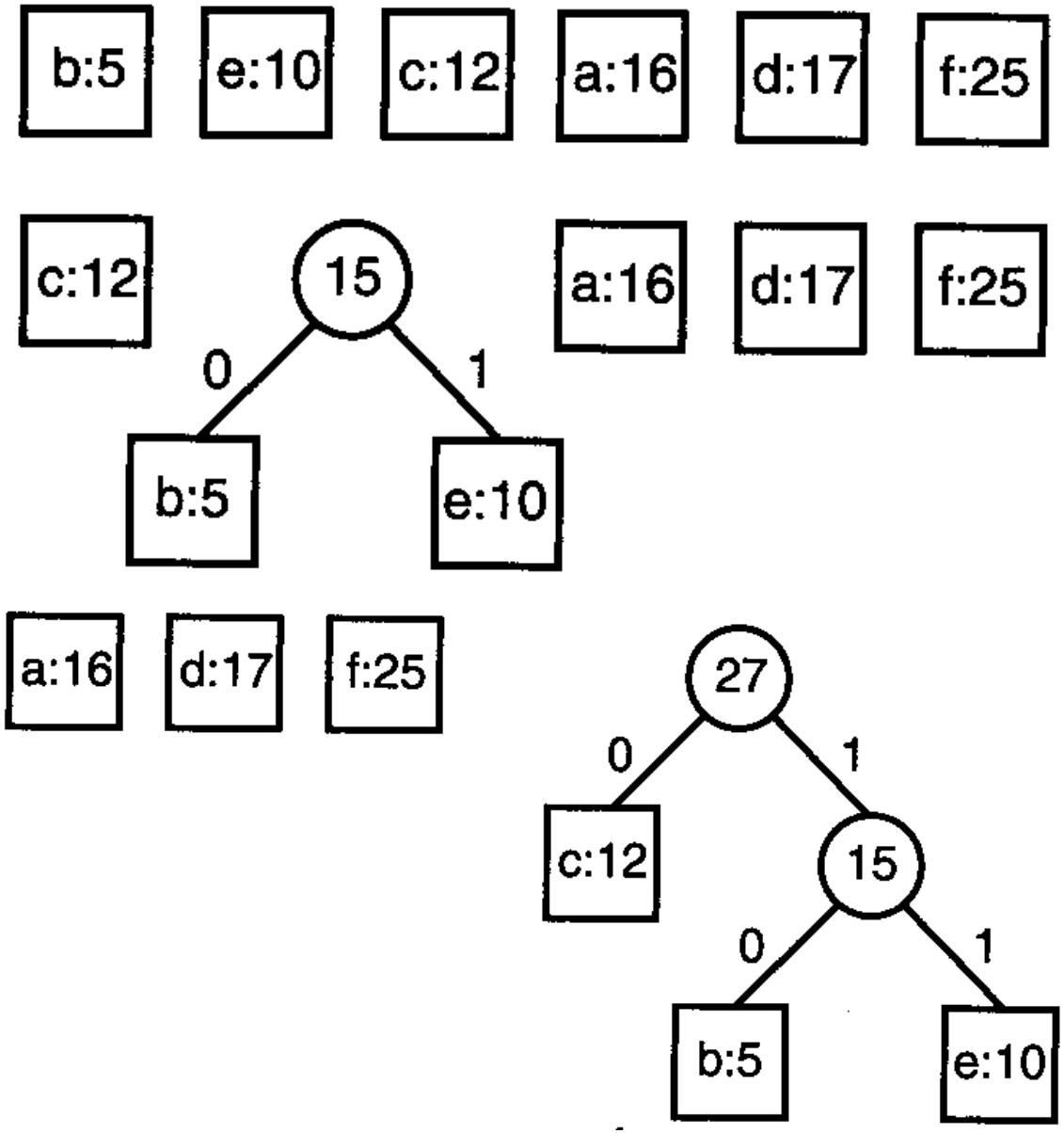
FIGURE 8.5 Functional block diagram of a general image compression system.

کدگذاری هافمن

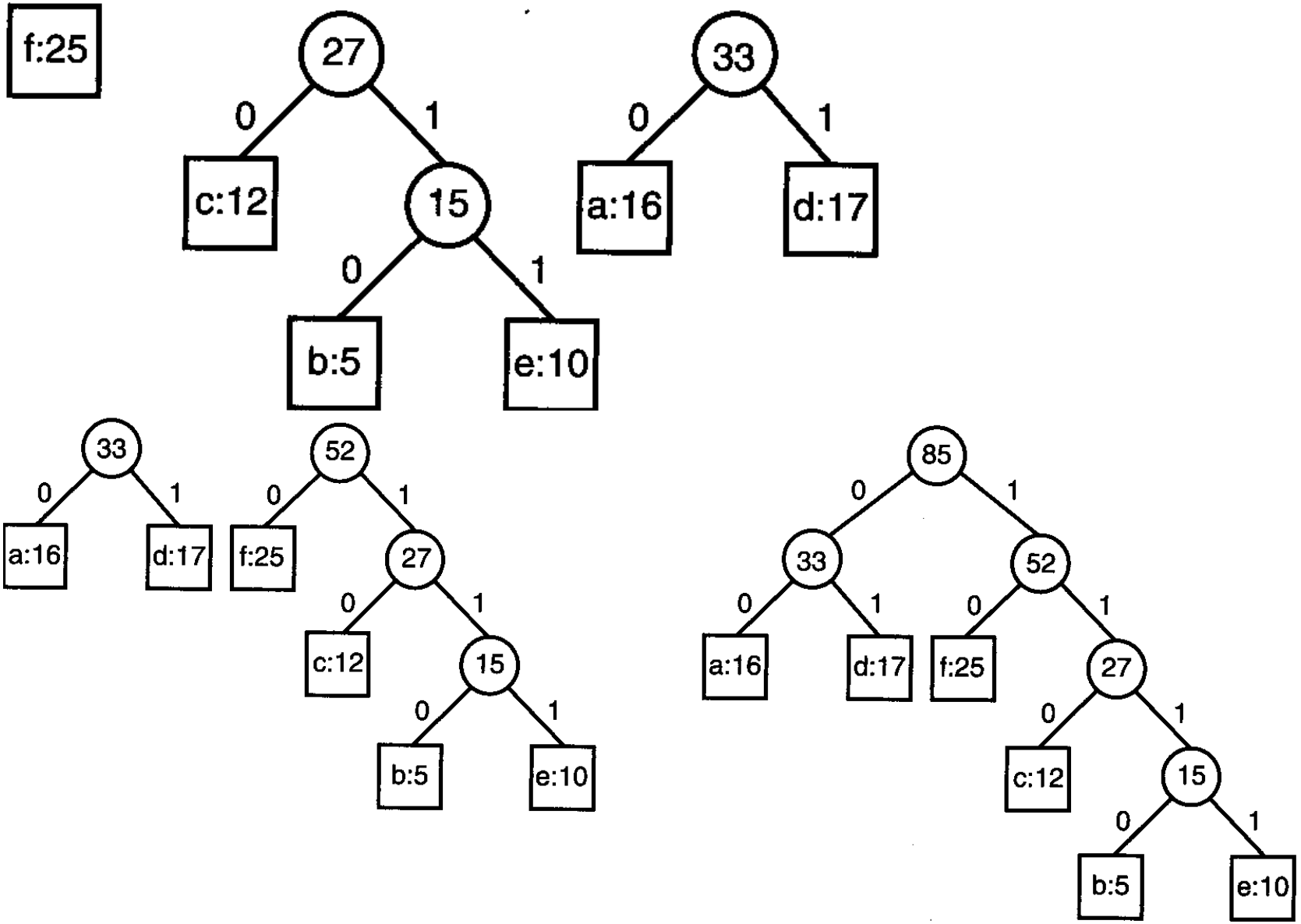
Huffman Encoding

- کدگذاری طول متغیر
- کاهش افزونگی کدگذاری
- مورد استفاده در
  - JPEG
  - MPEG-1,2,4
  - H.264،H.263،H.262،H.261
- اگر نمادها به صورت تک تک کد شوند این روش بهینه است.
- به همراه کد باید درخت نیز ارسال شود.
- بدون اتلاف

Huffman Encoding



Huffman Encoding …



کدگذاری مبتنی بر دیکشنری
LZW encoding- (Lempel–Ziv–Welch)

- روشی در راستای حذف افزونگی بین پیکسلی
- این روش در کدگذار و کدگشا هنگام دریافت داده، دیکشنری یکسانی را به صورت پویا تولید می‌کند.
- فشرده سازی بدون اتلاف

LZW Encoding

برای حذف افزنگی بین پیکسلی

```
BEGIN
  s = next input character;
  while not EOF {
      c = next input character;
        if    s + c exists in the dictionary
                  s = s + c;
      else
       {   output the code for s;
           add string s+c to the dictionary with a
new code;
           s = c;       }
                      }
  output the code for s;
END
```

کدگذاری بر پایه دیکشنری(دیکشنری اولیه)

Code	String
1	A
2	B
3	C

- مثال: کدکردن رشته "ABABBABCABABBA"

کدگذاری بر پایه دیکشنری(مثال)

“ABABBABCABABBA”

خروجی: 1 2 4 5 2 3 4 6 1.

به جای فرستادن ۱۴ کاراکتر تنها لازم است ۹ کد فرستاده شود.

نرخ فشرده‌سازی:

14/9 = 1.56

S	C	Output	Code	String
			1	A
			2	B
			3	C
A	B	1	4	AB
B	A	2	5	BA
A	B			
AB	B	4	6	ABB
B	A			
BA	B	5	7	BAB
B	C	2	8	BC
C	A	3	9	CA
A	B			
AB	A	4	10	ABA
A	B			
AB	B			
ABB	A	6	11	ABBA
A	EOF	1		

```
s = next input character;
while not EOF
 {
  c = next input character;
  if    s + c exists in the dictionary
                    s = s + c;
  else
       {
     output the code for s;
     add string s+c to the dic.
       with a new code;
     s = c;
    }
 }
output the code for s;
```

کدگشایی بر پایه دیکشنری

- **مثال : رمزگشایی** برای رشته "ABABBABCABABBA" .
کدهای ورودی به decoder: 1 2 4 5 2 3 4 6 1

S	K	Entry/output	Code	String
			1	A
			2	B
			3	C
NIL	1	A		
A	2	B	4	AB
B	4	AB	5	BA
AB	5	BA	6	ABB
BA	2	B	7	BAB
B	3	C	8	BC
C	4	AB	9	CA
AB	6	ABB	10	ABA
ABB	1	A	11	ABBA
A	EOF			

```
BEGIN
    s = NIL;
    while not EOF
    {
            k = next input code;
            entry = dictionary
entry for k;
            output entry;
            if (s != NIL)
            add string s+
entry[0] to dictionary with a
new code;
            s = entry;
    }
END
```

تصویر ورودی

39	39	126	126
39	39	126	126
39	39	126	126
39	39	126	126

TABLE 8.7 LZW coding example.

Currently Recognized Sequence	Pixel Being Processed	Encoded Output	Dictionary Location (Code Word)	Dictionary Entry
	39			
39	39	39	256	39-39
39	126	39	257	39-126
126	126	126	258	126-126
126	39	126	259	126-39
39	39			
39-39	126	256	260	39-39-126
126	126			
126-126	39	258	261	126-126-39
39	39			
39-39	126			
39-39-126	126	260	262	39-39-126-126
126	39			
126-39	39	259	263	126-39-39
39	126			
39-126	126	257	264	39-126-126
126		126		

کد هافمن

بخشی از اطلاعات مربوط به یک کد هافمن مطابق شکل است. بیت‌های x,y,z,w,p,q را تعیین کنید. (۱۵)

کلمه کد	کاراکتر
xy	A
xzw	B
$xzpq$	C
1111	D
01	E