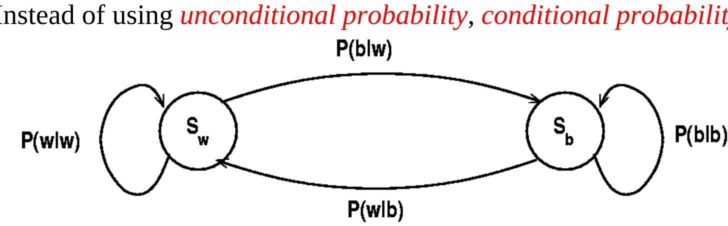
Run-length Encoding

Computer Science Department
CS4481b/9628b: Image Compression
Winter 2017
Instructor: Mahmoud R. El-Sakka
Office: MC-419
Email: elsakka@csd.uwo.ca
Phone: 519-661-2111 x86996

Run-length Encoding

- Perhaps the facsimile, or simply the fax, compression is one of the earliest applications of lossless compression in the modern era
- The word *facsimile* comes from the Latin (***fac simile***), which means ***make it the same!***

- In facsimile transmission:
  - A page is scanned and converted into a sequence of black and white pixels
  - These binary pixel values are encoded and transmitted
  - Instead of using *unconditional probability*, *conditional probability* is used



- ***Which probabilities are going to dominant?***

One-dimensional Run-length Encoding

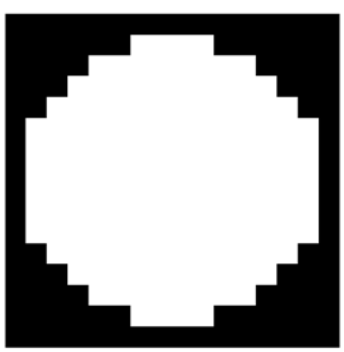
- One-dimensional RL encoding represents each image row by a sequence of lengths that describes successive runs of black and white pixels; where

  P(w|w) > P(b|w) and ***P(b|b) > P(w|b)***

- Basic concept
  - Assume that each row begins with a white (or a black) run;
  - In a left to right raster scan fashion, encode each contiguous group of 0's, or 1's, by its length
  - Whenever you have a row starts with a black (or a white) run, the length of this run will be zero

One-dimensional Run-length Encoding

- Example: Encode the shown image, using one-dimensional run-length encoding. Assuming that the first run of each row is black.



[16], [6, 4, 6], [4, 8, 4], [3, 10, 3],

[2, 12, 2], [1, 14, 1], [1, 14, 1], [1, 14, 1],

[1, 14, 1], [1, 14, 1], [1, 14, 1], [2, 12, 2],

[3, 10, 3], [4, 8, 4], [6, 4, 6], and [16]

- The white runs are: 4, 8, 10, 12, and 14
- The black runs are: 1, 2, 3, 4, 6, and 16
- ***Do we need to encode "[" and "]"? Why?***

One-dimensional Run-length Encoding

- Additional compression can be realized by Huffman to encode the run length values themselves;
  In such case, two encoders would be used. ***WHY two?***
- ***Exercise***: Calculate the number of required bits to encode the run length values in the previous example using:
  - Fixed length codeword encoding for
    - both the black and the white together
    - each of the black and the white separately
  - Huffman encoding for
    - both the black and the white together
    - each of the black and the white separately
- ***How can we extend this scheme to use it in compressing gray-scale images?***

Topic 09: Run-length Encoding

One-dimensional Run-length Encoding

- To keep the size of the Huffman table reasonably small,
  - Codes are defined for run lengths from 0 to 63 and then for 64, 128, 192, and 1728 (1728 pixels per line for A4 paper with 204 dpi)
  - Values of 64 or greater are encoded with a two-part codeword
- Encoded rows of pixels are terminated by a special end-of-line codeword to force codeword synchronization (resetting errors, if any)
- This codeword encoding is called Modified Huffman (MH) method

One-dimensional Run-length Encoding

White run length	Code word	Black run length	Code word
0	00110101	0	0000110111
1	000111	1	010
2	0111	2	11
3	1000	3	10
4	1011	4	011
5	1100	5	0011
6	1110	6	0010
7	1111	7	00011
8	10011	8	000101
9	10100	9	000100
10	00111	10	0000100
11	01000	11	0000101
12	001000	12	0000111
13	000011	13	00000100
14	110100	14	00000111
15	110101	15	000011000

White run length	Code word	Black run length	Code word
16	101010	16	0000010111
17	101011	17	0000011000
18	0100111	18	0000001000
19	0001100	19	00001100111
20	0001000	20	00001101000
21	0010111	21	00001101100
22	0000011	22	00000110111
23	0000100	23	00000101000
24	0101000	24	00000010111
25	0101011	25	00000011000
26	0010011	26	000011001010
27	0100100	27	000011001011
28	0011000	28	000011001100
29	00000010	29	000011001101
30	00000011	30	000001101000
31	00011010	31	000001101001

One-dimensional Run-length Encoding

White run length	Code word	Black run length	Code word
32	00011011	32	000001101010
33	00010010	33	000001101011
34	00010011	34	000011010010
35	00010100	35	000011010011
36	00010101	36	000011010100
37	00010110	37	000011010101
38	00010111	38	000011010110
39	00101000	39	000011010111
40	00101001	40	000001101100
41	00101010	41	000001101101
42	00101011	42	000011011010
43	00101100	43	000011011011
44	00101101	44	000001010100
45	00000100	45	000001010101
46	00000101	46	000001010110
47	00001010	47	000001010111

White run length	Code word	Black run length	Code word
48	00001011	48	000001100100
49	01010010	49	000001100101
50	01010011	50	000001010010
51	01010100	51	000001010011
52	01010101	52	000000100100
53	00100100	53	000000110111
54	00100101	54	000000111000
55	01011000	55	000000100111
56	01011001	56	000000101000
57	01011010	57	000001011000
58	01011011	58	000001011001
59	01001010	59	000000101011
60	01001011	60	000000101100
61	00110010	61	000001011010
62	00110011	62	000001100110
63	00110100	63	000001100111

One-dimensional Run-length Encoding

White run length	Code word	Black run length	Code word
64	11011	64	0000001111
128	10010	128	000011001000
192	010111	192	000011001001
256	0110111	256	000001011011
320	00110110	320	000000110011
384	00110111	384	000000110100
448	01100100	448	000000110101
512	01100101	512	0000001101100
576	01101000	576	0000001101101
640	01100111	640	0000001001010
704	011001100	704	0000001001011
768	011001101	768	0000001001100
832	011010010	832	0000001001101
896	011010011	896	0000001110010
960	011010100	960	0000001110011
1024	011010101	1024	0000001110100

White run length	Code word	Black run length	Code word
1088	011010110	1088	0000001110101
1152	011010111	1152	0000001110110
1216	011011000	1216	0000001110111
1280	011011001	1280	0000001010010
1344	011011010	1344	0000001010011
1408	011011011	1408	0000001010100
1472	010011000	1472	0000001010101
1536	010011001	1536	0000001011010
1600	010011010	1600	0000001011011
1664	011000	1664	0000001100100
1728	010011011	1728	0000001100101
EOL	000000000001	EOL	000000000001

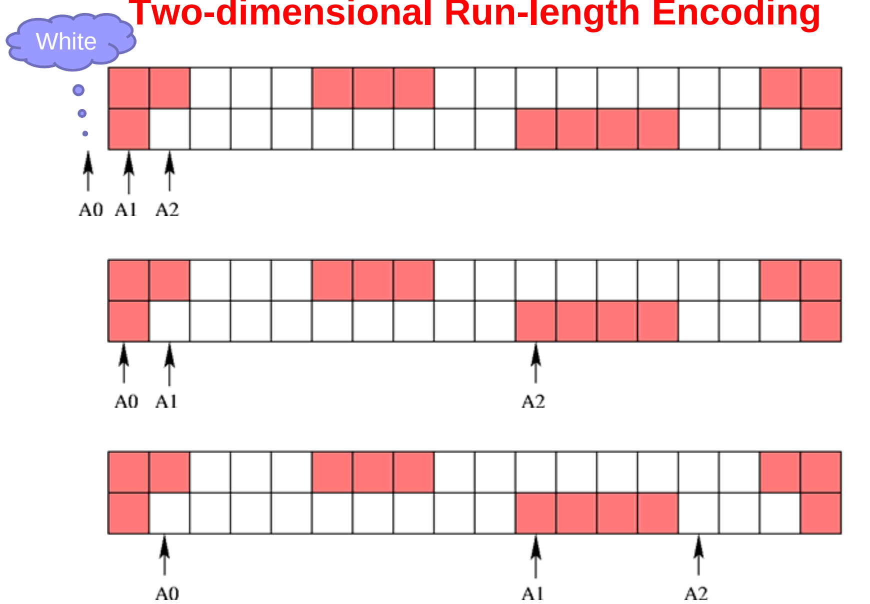
Two-dimensional Run-length Encoding

- One-dimensional run-length encoding concept can be easily extended to two-dimension
- The basic idea is to encode the starting position of a run in the current line relative to the previous line
- The RL version that is used in the CCITT group 3 and 4 standards is a *modification* of a two-dimensional encoding scheme called *Relative Element Address Designate* (READ); hence it is often referred to as *Modified READ* (MR)
- CCITT Group 3 standard (T.4) is published in 1980, whereas CCITT Group 4 standard (T.6) is published in 1984

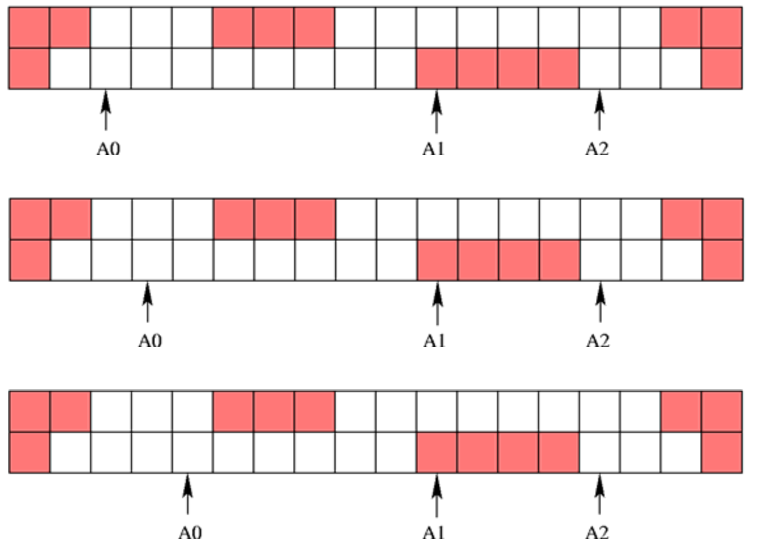
Two-dimensional Run-length Encoding

- To understand the two-dimensional RL encoding, consider the following definitions of ***A0, A1, A2, B1, B2***
  - ***A0***: The last pixel whose value is ***known to both encoder and decoder***; At the beginning of each line, ***A0*** refers to an *imaginary* white pixel to the *left* of the first actual pixel
  - ***A1***: The first transition pixel to the right of ***A0***; its color should be the ***opposite*** to the ***A0*** color; ***known only to the encoder***
  - ***A2***: The second transition pixel to the right of ***A0***; its color should be the ***same*** as the ***A0*** color; ***known only to the encoder***
- ***A0*** can refer to an *imaginary* pixel to the *left* of the first actual pixel (initial condition), but not ***A1*** or ***A2***
- Note: the *left imaginary* pixel can be assumed as white (or black)
- ***A1*** and ***A2*** can refer to an *imaginary* pixel to the *right* of the last actual pixel, but not ***A0***
- We assume that this *right imaginary* pixel has color that is ***opposite*** to the last actual pixel

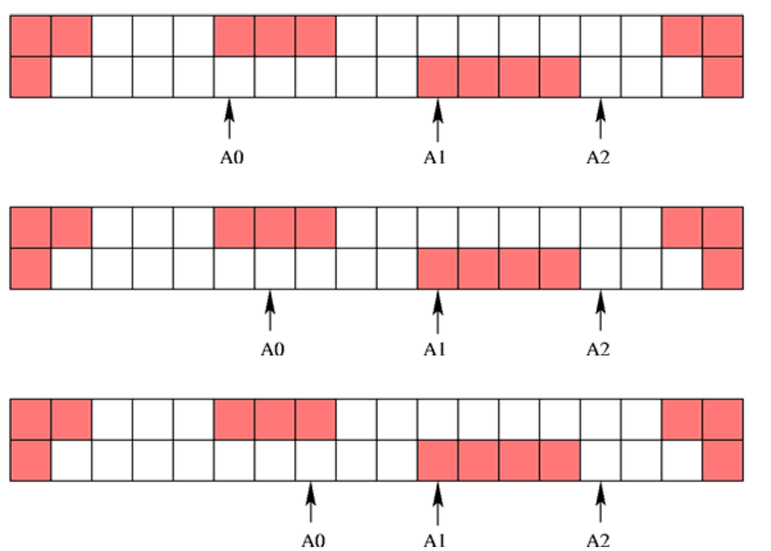
Two-dimensional Run-length Encoding



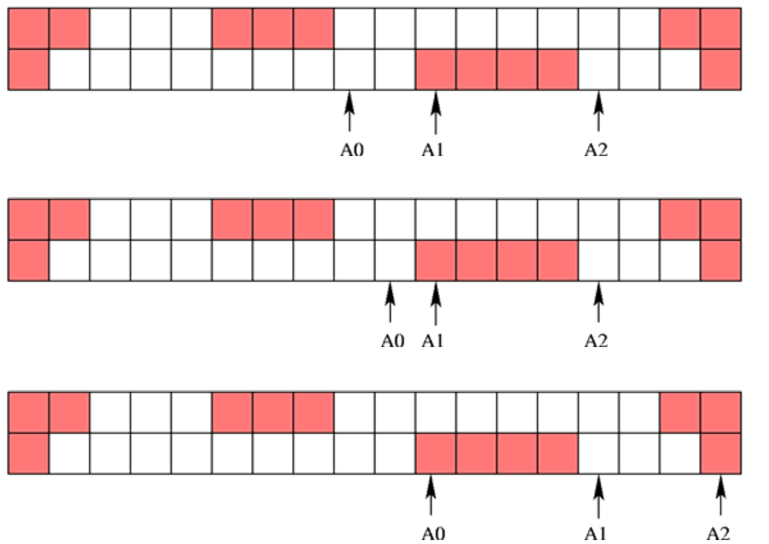
Two-dimensional Run-length Encoding



Two-dimensional Run-length Encoding

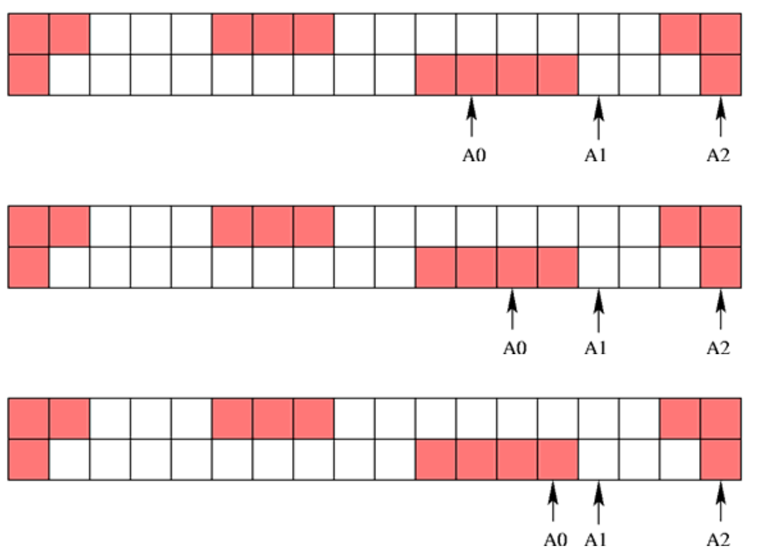


Two-dimensional Run-length Encoding

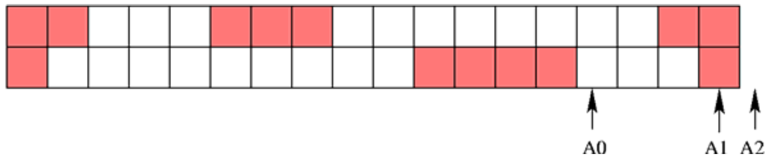


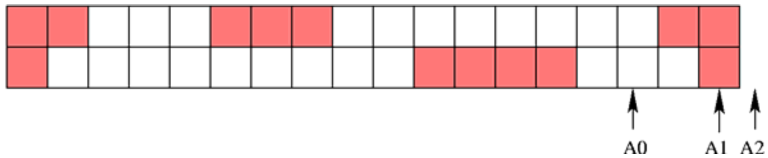
Topic 09: Run-length Encoding

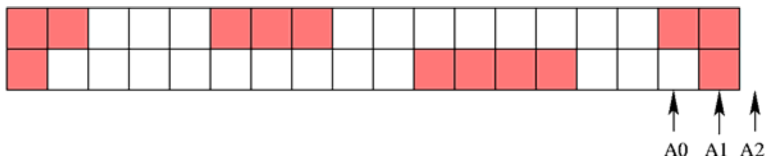
Two-dimensional Run-length Encoding



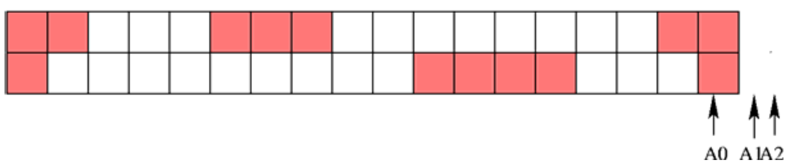
Two-dimensional Run-length Encoding







Two-dimensional Run-length Encoding

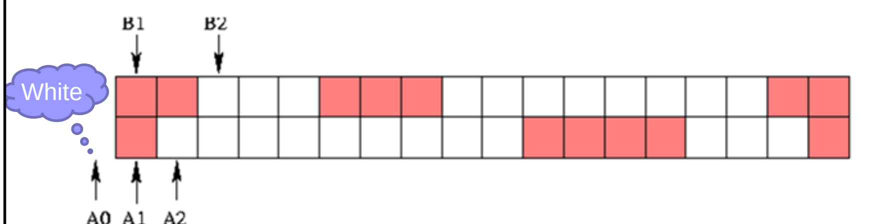


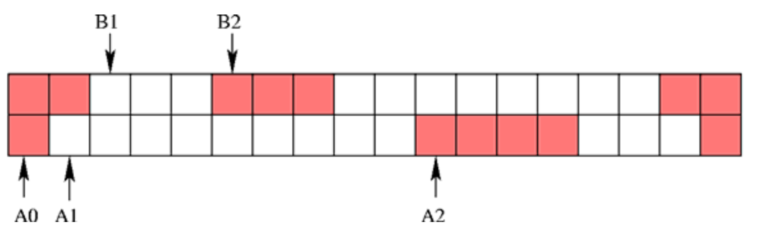
Two-dimensional Run-length Encoding

- *B1*: The *first transition pixel* on the line above the line currently being encoded *to the right* of *A0* whose color is the *opposite* to *A0*; *known to both encoder and decoder*
- *B2*: The first transition pixel to the right of *B1* in the line above the line currently being encoded; its color should be the *same* as the *A0* color; *known to both encoder and decoder*

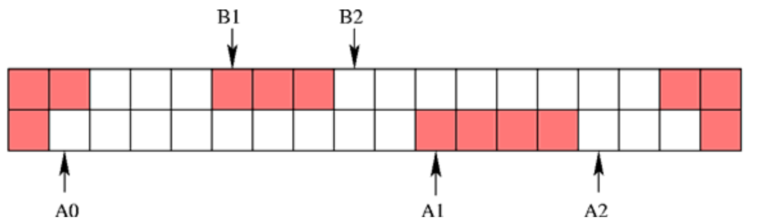
- *B1* and *B2* can refer to an *imaginary pixel* to the *right* of the last actual pixel

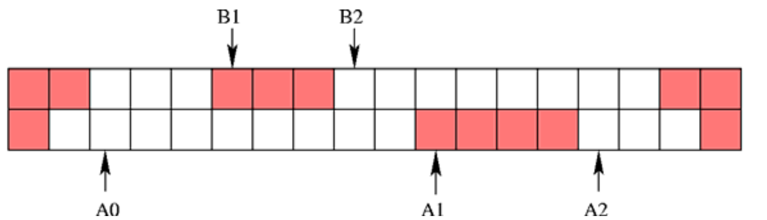
Two-dimensional Run-length Encoding





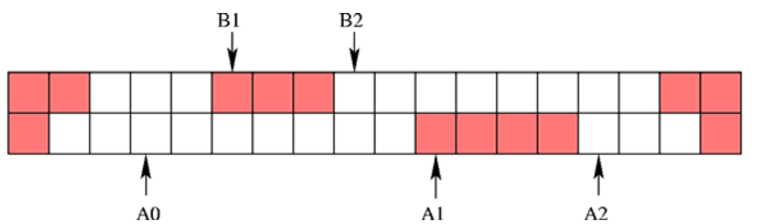
Two-dimensional Run-length Encoding

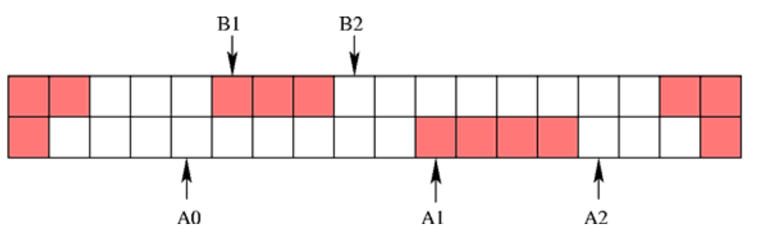




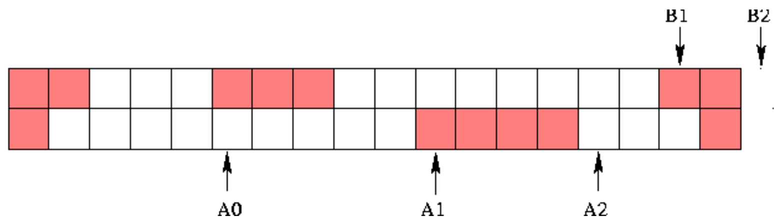
Topic 09: Run-length Encoding

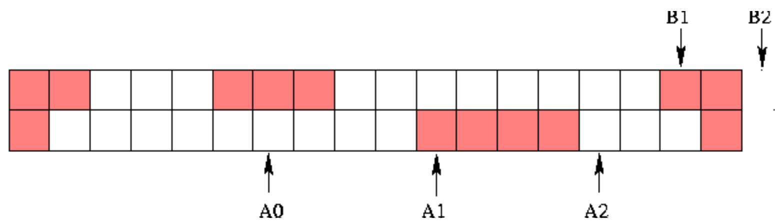
Two-dimensional Run-length Encoding



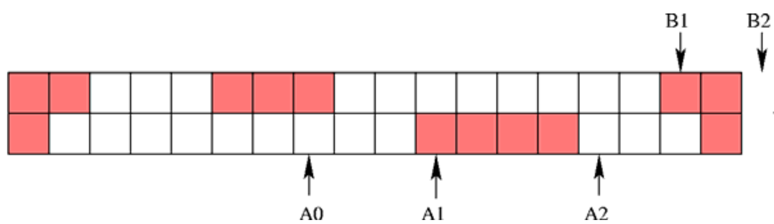


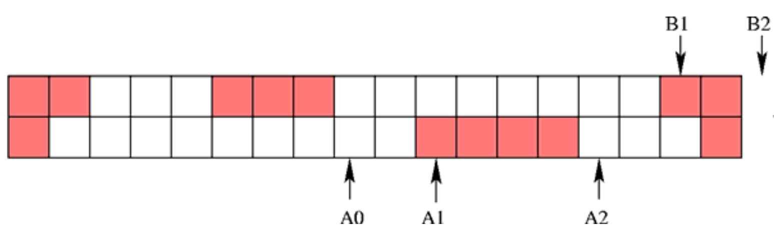
Two-dimensional Run-length Encoding



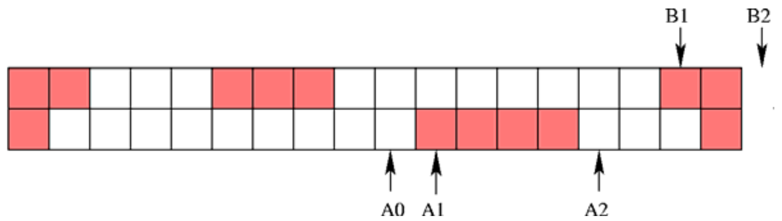


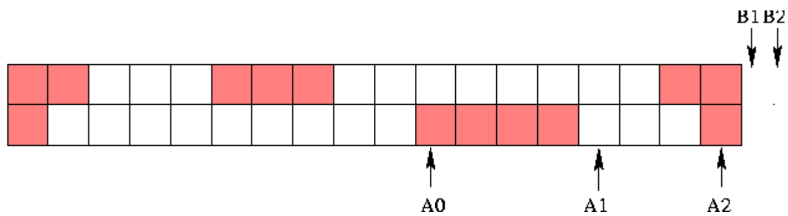
Two-dimensional Run-length Encoding



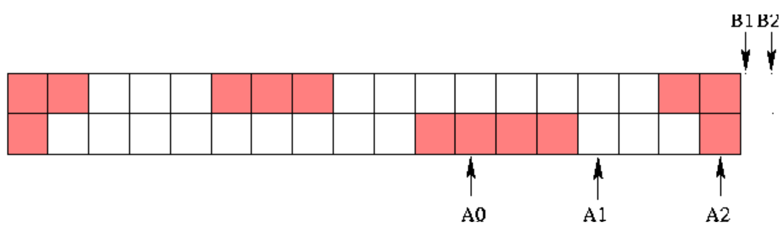


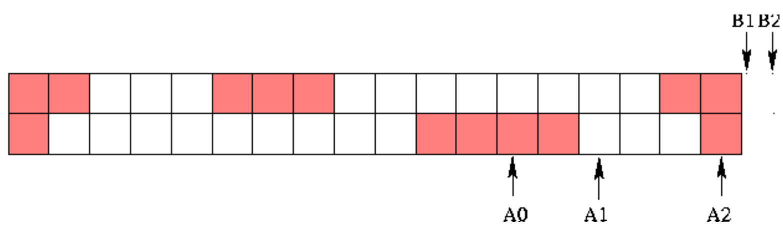
Two-dimensional Run-length Encoding



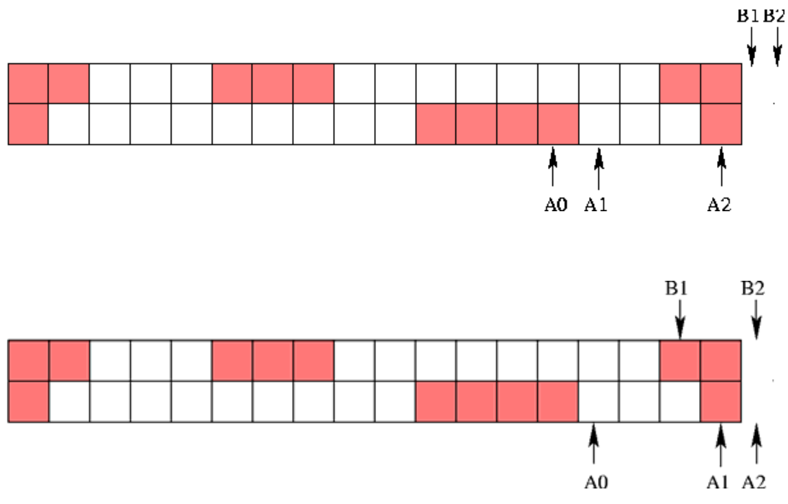


Two-dimensional Run-length Encoding

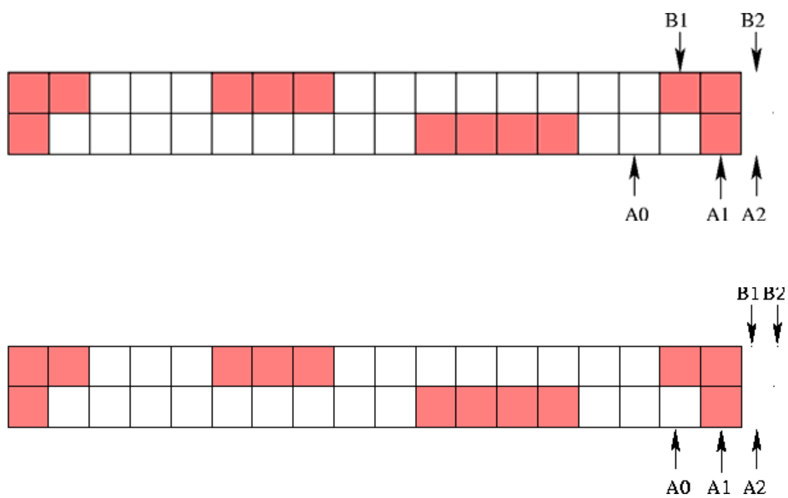




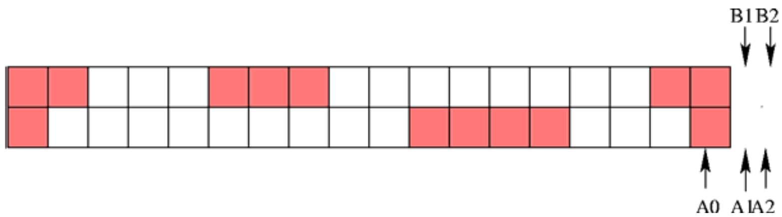
Two-dimensional Run-length Encoding



Two-dimensional Run-length Encoding



Two-dimensional Run-length Encoding

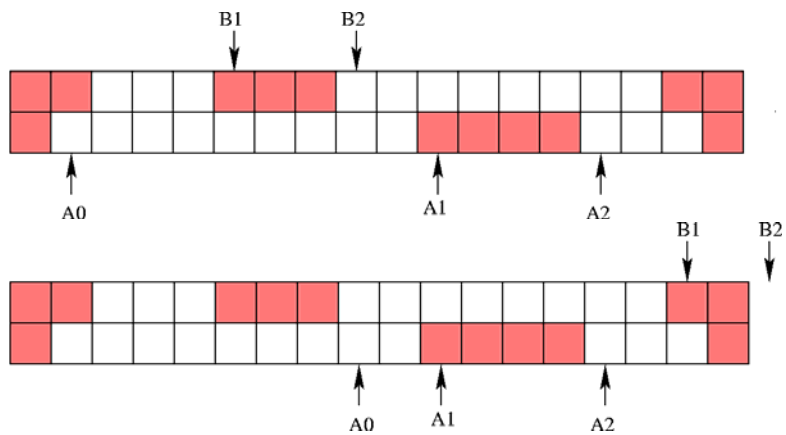


Two-dimensional Run-length Encoding

- There are three modes in the two-dimensional run-length encoding scheme:
  - ***Pass mode***
  - ***Vertical mode***
  - ***Horizontal mode***

Two-dimensional Run-length Encoding

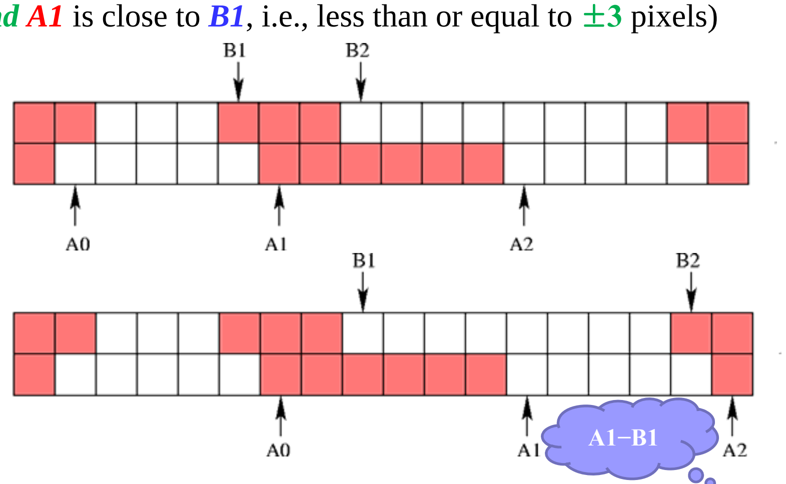
- ***Pass mode*** (When ***B1*** and ***B2*** lie between ***A0*** and ***A1***)



The generated code should be: <pass mode>

Two-dimensional Run-length Encoding

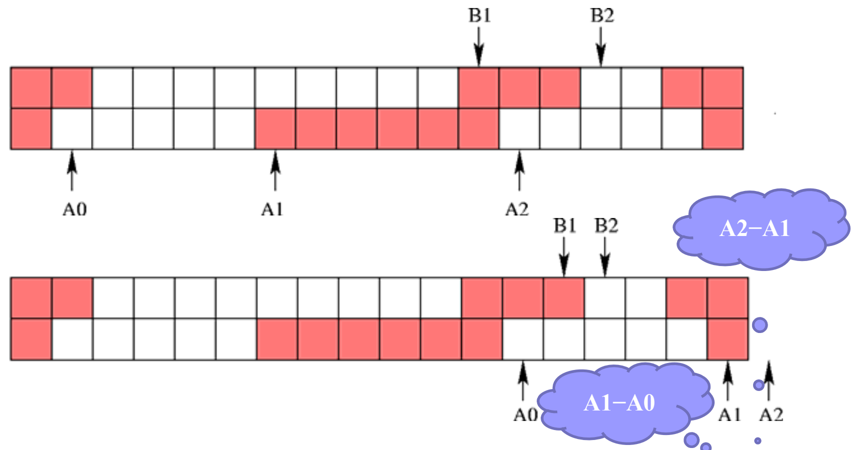
- ***Vertical mode*** (When ***B1*** and ***B2*** do not lie between ***A0*** and ***A1***, ***and A1*** is close to ***B1***, i.e., less than or equal to ±3 pixels)



The generated code should be: <Vertical mode: +1>

Two-dimensional Run-length Encoding

- ***Horizontal mode*** (When ***B1*** and ***B2*** do not lie between ***A0*** and ***A1***, ***and A1*** is far away from ***B1***, i.e., more than **3** pixels)



The generated code should be: <Horizontal mode: 5, 6>

Two-dimensional Run-length Encoding

- The encoding process continues until the entire row is encoded
- ***Exercise***: Continue the encoding process of the last three examples until you finish encoding the entire row. I.e.,
  - you keep
    - determining the *correct mode* and
    - relocating the ***A0***, ***A1***, ***A2***, ***B1***, and ***B2*** locations;
  - until you encode the entire row

Two-dimensional Run-length Encoding

- As an initial step, one-dimensional run-length encoding is applied first, followed by two-dimensional run-length encoding
- In general, two-dimensional run-length encoding achieves higher compression than that in the one-dimensional run-length encoding, since it utilizes the vertical correlation in an image as well

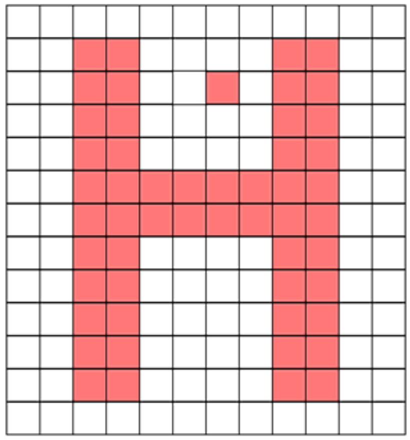
Topic 09: Run-length Encoding

Two-dimensional Run-length Encoding

- Since the encoding of a line in the two-dimensional run-length encoding algorithm is based on the previous line, an error in one line could propagate to all other lines in the transmission
- To prevent this from happening, a recommendation is made to require that after each line is encoded with the one-dimensional run-length encoding algorithm, at most ***K-1*** lines will be encoded using the two-dimensional run-length encoding algorithm, where
  - ***K=2*** in the standard resolution images (*i.e., **one** line to utilize 1D-RL and **one** line to utilize 2D-RL*) and
  - ***K=4*** in the high resolution images (*i.e., **one** line to utilize 1D-RL and **three** line to utilize 2D-RL*)

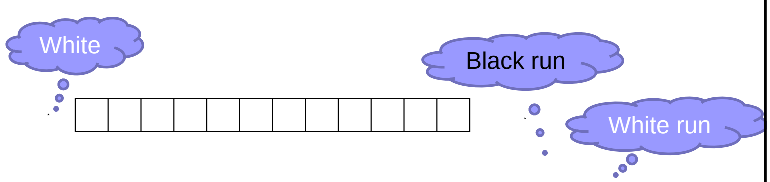
Two-dimensional Run-length Encoding

- ***Example***: Encode the following image. Consider it as a high resolution image. Assume that the *left imaginary* pixel is white.



Two-dimensional Run-length Encoding

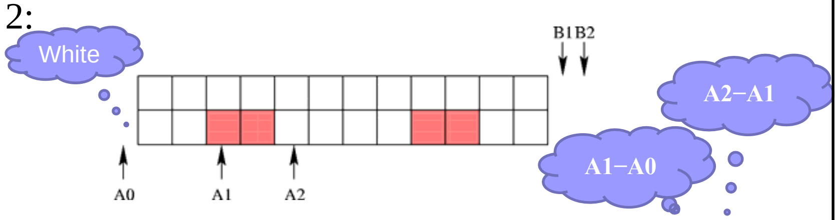
- Row 1:



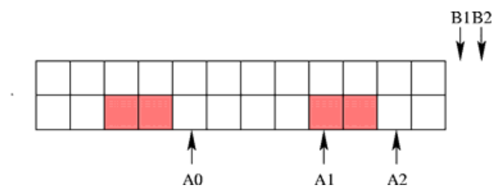
The generated code should be: <RL-1D: 0, 12>

Two-dimensional Run-length Encoding

- Row 2:



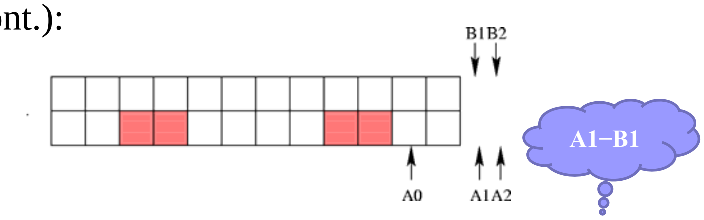
The generated code should be: <Horizontal mode: 3, 2>



The generated code should be: <Horizontal mode: 4, 2>

Two-dimensional Run-length Encoding

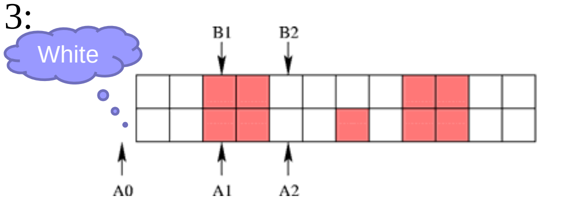
- Row 2 (Cont.):



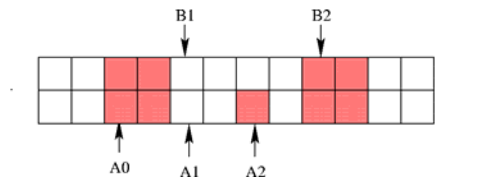
The generated code should be: <Vertical mode: 0>

Two-dimensional Run-length Encoding

- Row 3:



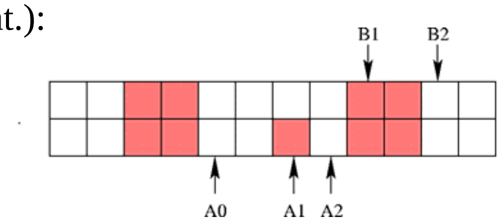
The generated code should be: <Vertical mode: 0>



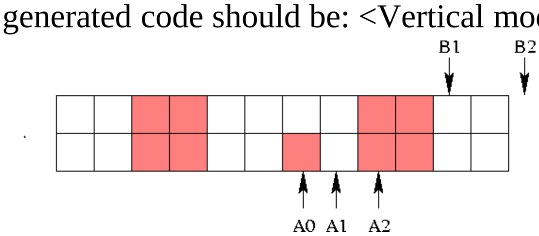
The generated code should be: <Vertical mode: 0>

Two-dimensional Run-length Encoding

- Row 3 (cont.):



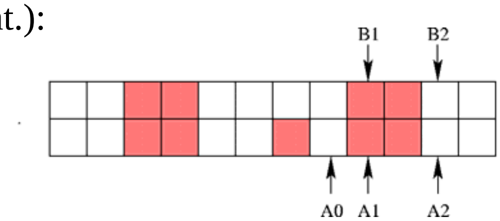
The generated code should be: <Vertical mode: -2>



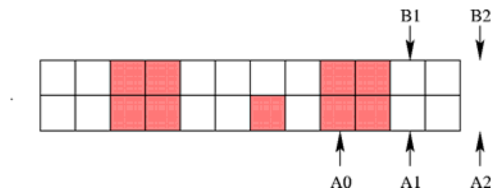
The generated code should be: <Vertical mode: -3>

Two-dimensional Run-length Encoding

- Row 3 (cont.):



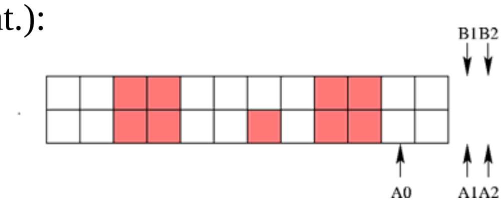
The generated code should be: <Vertical mode: 0>



The generated code should be: <Vertical mode: 0>

Two-dimensional Run-length Encoding

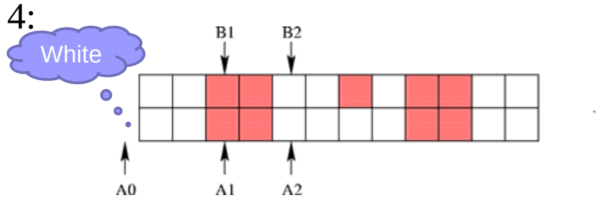
- Row 3 (cont.):



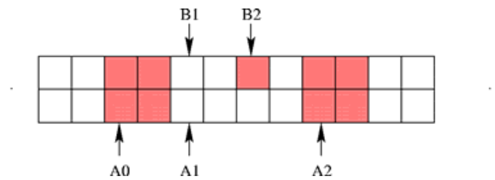
The generated code should be: <Vertical mode: 0>

Two-dimensional Run-length Encoding

- Row 4:



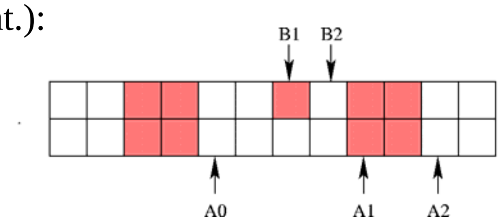
The generated code should be: <Vertical mode: 0>



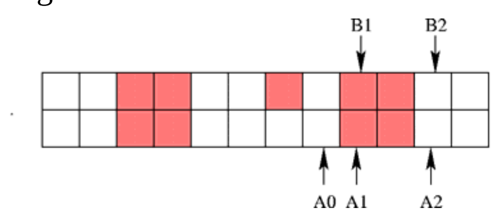
The generated code should be: <Vertical mode: 0>

Two-dimensional Run-length Encoding

- Row 4 (cont.):



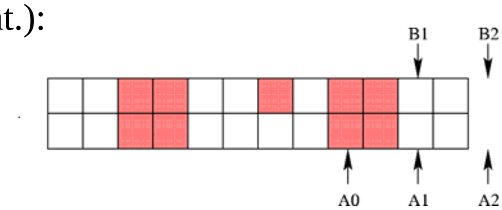
The generated code should be: <Pass mode>



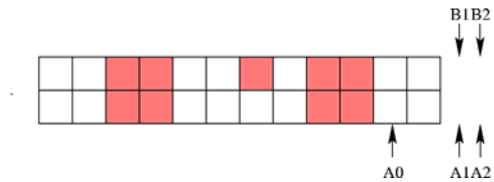
The generated code should be: <Vertical mode: 0>

Two-dimensional Run-length Encoding

- Row 4 (cont.):



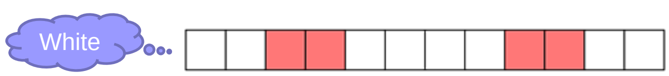
The generated code should be: <Vertical mode: 0>



The generated code should be: <Vertical mode: 0>

Two-dimensional Run-length Encoding

- Row 5:



The generated code should be: <RL-1D: 0, 2, 2, 4, 2, 2>

Two-dimensional Run-length Encoding

- Decode the following 2D-RL compressed binary image.

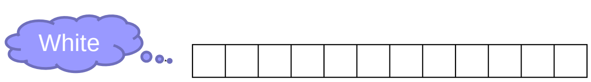
  <IMAGE Width: 12>; <RL-1D: 0, 12>; <Horizontal mode: 3, 2>;
  <Horizontal mode: 4, 2>; <Vertical mode: 0>; <Vertical mode: 0>;
  <Vertical mode: 0>; <Vertical mode: -2>; <Vertical mode: -3>;
  <Vertical mode: 0>; <Vertical mode: 0>; <Vertical mode: 0>;
  <Vertical mode: 0>; <Vertical mode: 0>; <Pass mode>;
  <Vertical mode: 0>; <Vertical mode: 0>; <Vertical mode: 0>;
  <RL-1D: 0, 2, 2, 4, 2, 2>

Two-dimensional Run-length Decoding

- <IMAGE Width: 12>;

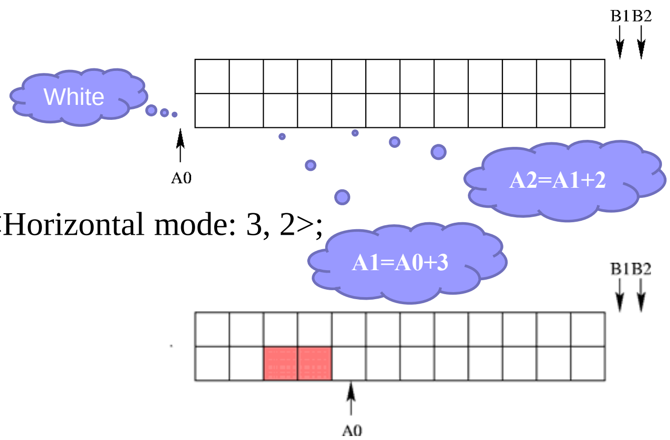
- Row 1:

  <RL-1D: 0, 12>;



Two-dimensional Run-length Decoding

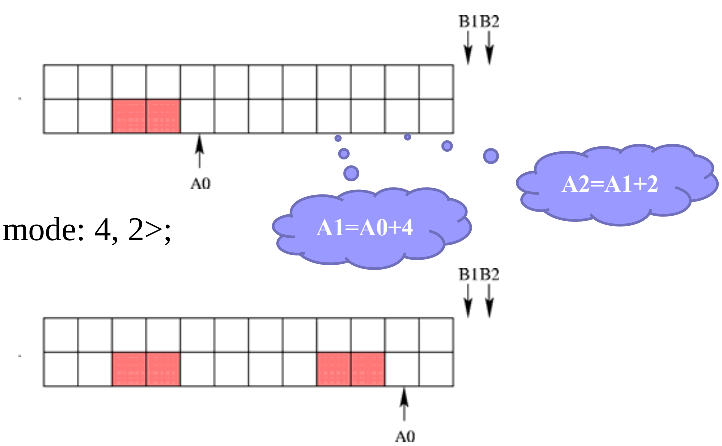
- Row 2:



<Horizontal mode: 3, 2>;

Two-dimensional Run-length Decoding

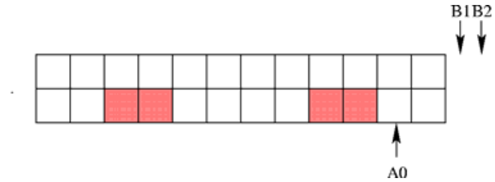
- Row 2 (cont.):



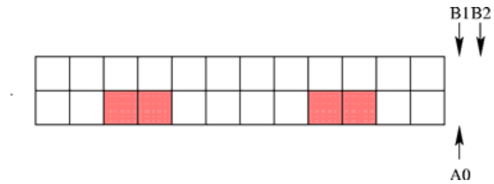
<Horizontal mode: 4, 2>;

Two-dimensional Run-length Decoding

- Row 2 (cont.):

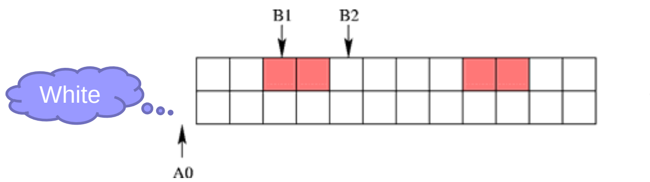


<Vertical mode: 0>;

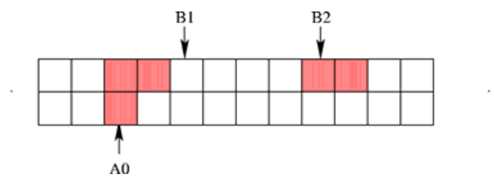


Two-dimensional Run-length Decoding

- Row 3:

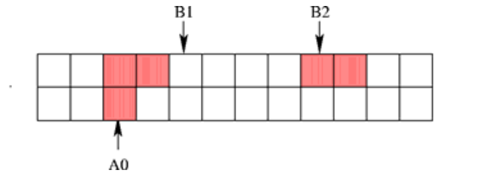


<Vertical mode: 0>;

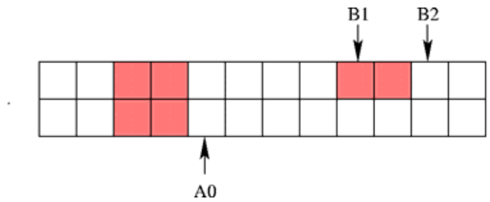


Two-dimensional Run-length Decoding

- Row 3 (cont.):

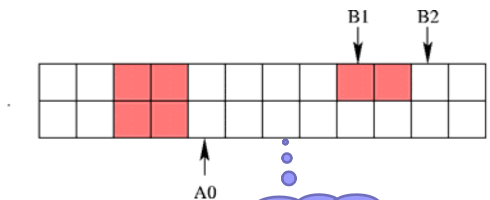


<Vertical mode: 0>;

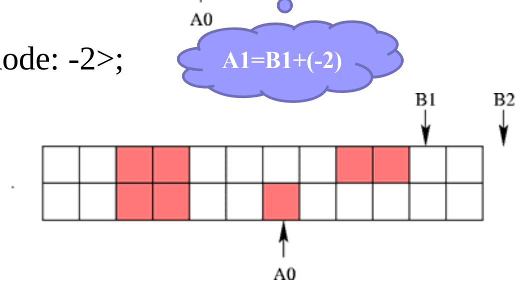


Two-dimensional Run-length Decoding

- Row 3 (cont.):

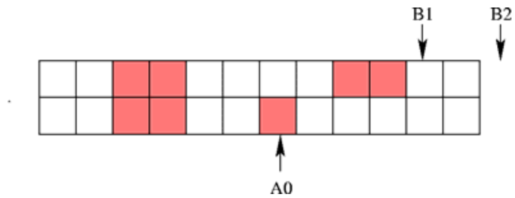


<Vertical mode: -2>;

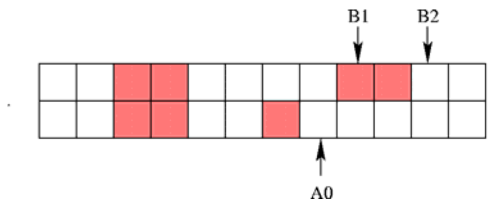


Two-dimensional Run-length Decoding

- Row 3 (cont.):

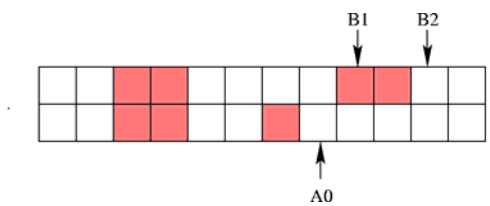


<Vertical mode: -3>;

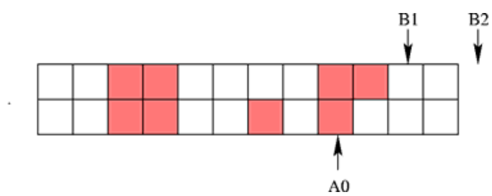


Two-dimensional Run-length Decoding

- Row 3 (cont.):

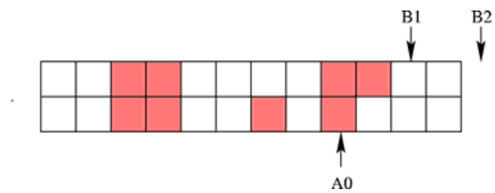


<Vertical mode: 0>;

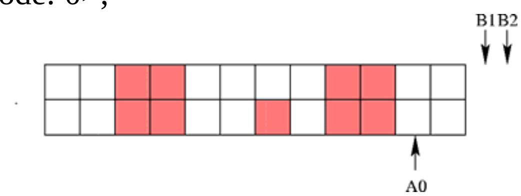


Two-dimensional Run-length Decoding

- Row 3 (cont.):



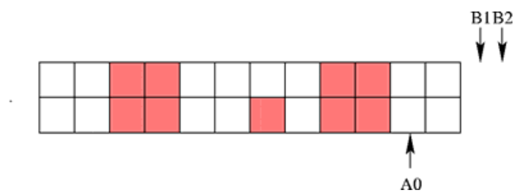
<Vertical mode: 0>;



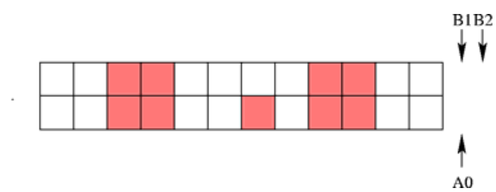
Topic 09: Run-length Encoding

Two-dimensional Run-length Decoding

- Row 3 (cont.):

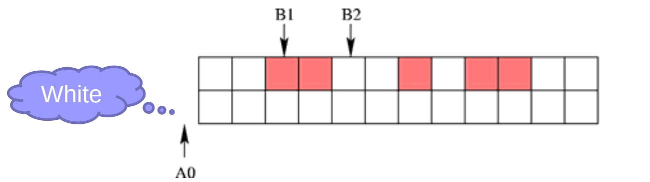


<Vertical mode: 0>;

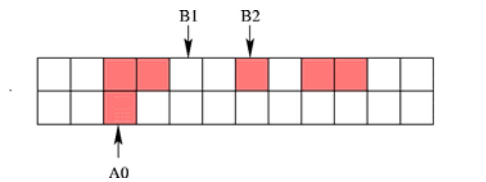


Two-dimensional Run-length Decoding

- Row 4:

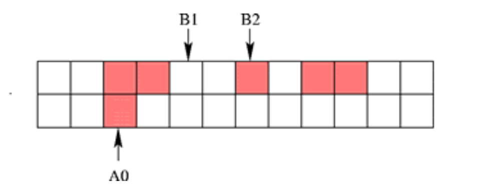


<Vertical mode: 0>;

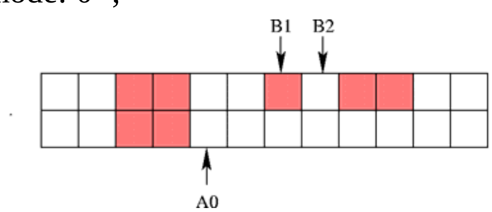


Two-dimensional Run-length Decoding

- Row 4 (cont.):

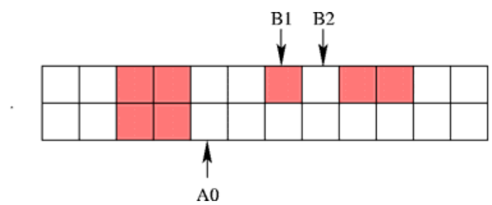


<Vertical mode: 0>;

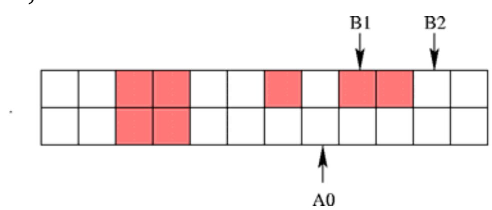


Two-dimensional Run-length Decoding

- Row 4 (cont.):

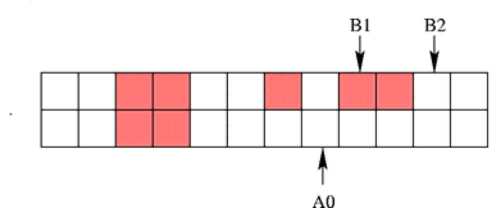


<Pass mode>;

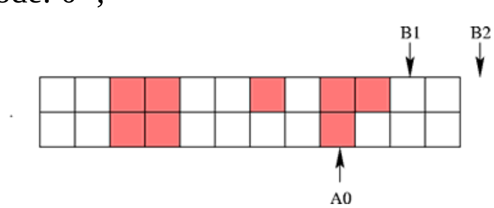


Two-dimensional Run-length Decoding

- Row 4 (cont.):

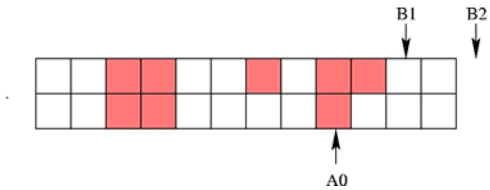


<Vertical mode: 0>;

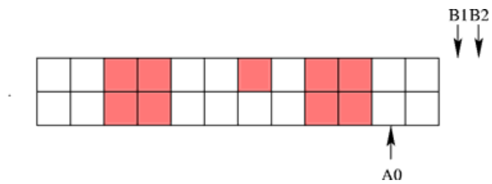


Two-dimensional Run-length Decoding

- Row 4 (cont.):



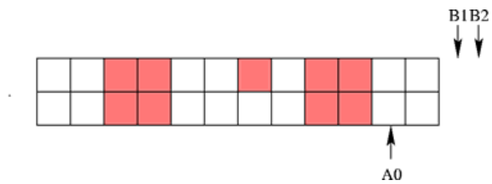
<Vertical mode: 0>;



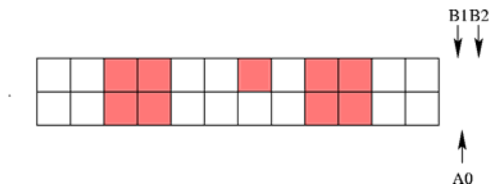
Topic 09: Run-length Encoding

Two-dimensional Run-length Decoding

- Row 4 (cont.):



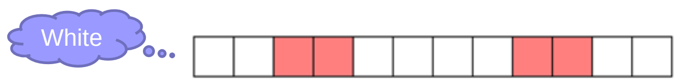
<Vertical mode: 0>;



Two-dimensional Run-length Decoding

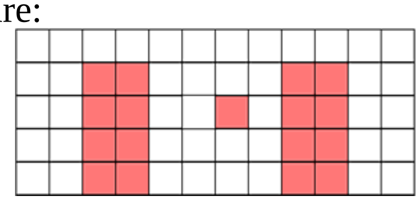
- Row 5:

<RL-1D: 0, 2, 2, 4, 2, 2>

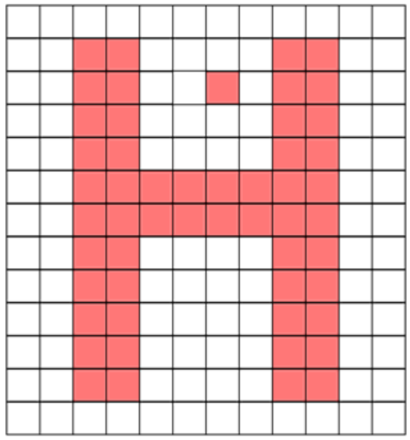


Two-dimensional Run-length Decoding

- Decoded picture:



Two-dimensional Run-length Decoding



- ***Exercise***: Continue encoding/decoding the rest of the image.

Binary Compression Performance

- Compression results highly depend on the image to be encoded
- One-dimensional RL encoding (i.e., MH only) *typically* achieve
  - 7:1 compression
- Two-dimensional RL encoding (Group 3) *typically* achieve
  - 9:1 compression (in case of normal resolution, i.e., k=2)
  - 11:1 compression (in case of normal resolution, i.e., k=4)
- Group 4 is just Group 3 compression without the need to go to the one-dimensional mode at all, even at the first line, where an *imaginary line* is assumed just before the first line to get things off the ground
- Two-dimensional RL encoding (Group 4) *typically* achieve
  - 14:1 compression