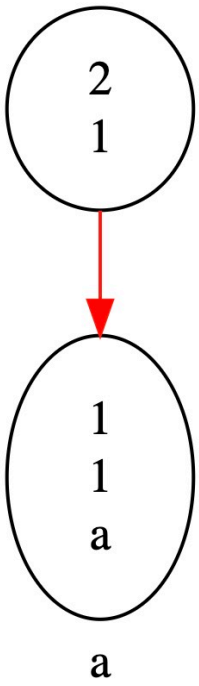
```
Java version used for this program is 12.0.2
HauffmanTest.java
============ a +++++++++++++++
'a' occurs  1 times
===Tree built in this order===============
Leaf    node  1: Character is 'a' Weight is  1
Transit node  2: Left a( 1) Right  ( 0) Weight =  1
You can see dot file at ./output/1.dot
==== Tree has 2 nodes =========
Original string cost = 7.0
Decoded  string cost = 1.0
% reduction = 85.71428571428571
```

```
digraph g {
  label = " a "
  "2\n1" -> "1\n1\na" [color=red]
}
```

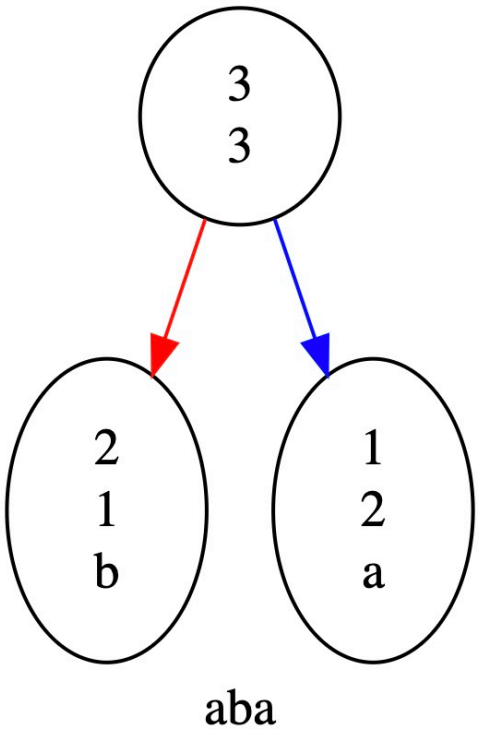
Engine: dot Format: svg



```
============ aba +++++++++++++++
'a' occurs  2 times
'b' occurs  1 times
===Tree built in this order===============
Leaf    node  1: Character is 'a' Weight is  2
Leaf    node  2: Character is 'b' Weight is  1
Transit node  3: Left b( 1) Right a( 2) Weight =  3
You can see dot file at ./output/2.dot
==== Tree has 3 nodes =========
Original string cost = 21.0
Decoded  string cost = 3.0
% reduction = 85.71428571428571
```

```
digraph g {
  label = " aba "
  "3\n3" -> "2\n1\nb" [color=red]
  "3\n3" -> "1\n2\na" [color=blue]
}
```

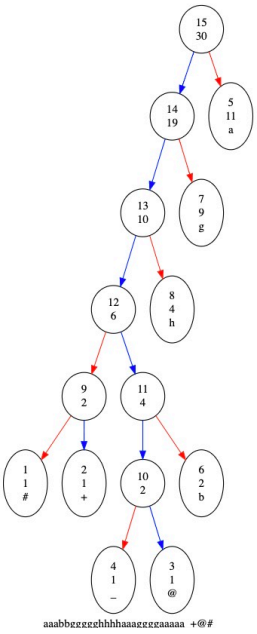
Engine: dot Format: svg



```
============ aaabbgggggghhhhaaaggggaaaaa_+@# +++++++++++++++
'#' occurs  1 times
'+' occurs  1 times
'@' occurs  1 times
'_' occurs  1 times
'a' occurs 11 times
'b' occurs  2 times
'g' occurs  9 times
'h' occurs  4 times
===Tree built in this order===============
Leaf    node  1: Character is '#' Weight is  1
Leaf    node  2: Character is '+' Weight is  1
Leaf    node  3: Character is '@' Weight is  1
Leaf    node  4: Character is '_' Weight is  1
Leaf    node  5: Character is 'a' Weight is 11
Leaf    node  6: Character is 'b' Weight is  2
Leaf    node  7: Character is 'g' Weight is  9
Leaf    node  8: Character is 'h' Weight is  4
Transit node  9: Left #( 1) Right +( 1) Weight =  2
Transit node 10: Left _( 1) Right @( 1) Weight =  2
Transit node 11: Left b( 2) Right  ( 2) Weight =  4
Transit node 12: Left  ( 2) Right  ( 4) Weight =  6
Transit node 13: Left h( 4) Right  ( 6) Weight = 10
Transit node 14: Left g( 9) Right  (10) Weight = 19
Transit node 15: Left a(11) Right  (19) Weight = 30
You can see dot file at ./output/3.dot
==== Tree has 15 nodes =========
Original string cost = 210.0
Decoded  string cost = 73.0
% reduction = 65.23809523809524
```

```
digraph g {
  label = " aaabbgggggghhhhaaaggggaaaaa_+@# "
  "9\n2" -> "1\n1\n#" [color=red]
  "9\n2" -> "2\n1\n+" [color=blue]
  "10\n2" -> "4\n1\n_" [color=red]
  "10\n2" -> "3\n1\n@" [color=blue]
  "11\n4" -> "6\n2\nb" [color=red]
  "11\n4" -> "10\n2" [color=blue]
  "12\n6" -> "9\n2" [color=red]
  "12\n6" -> "11\n4" [color=blue]
  "13\n10" -> "8\n4\nh" [color=red]
  "13\n10" -> "12\n6" [color=blue]
  "14\n19" -> "7\n9\ng" [color=red]
  "14\n19" -> "13\n10" [color=blue]
  "15\n30" -> "5\n11\na" [color=red]
  "15\n30" -> "14\n19" [color=blue]
}
```

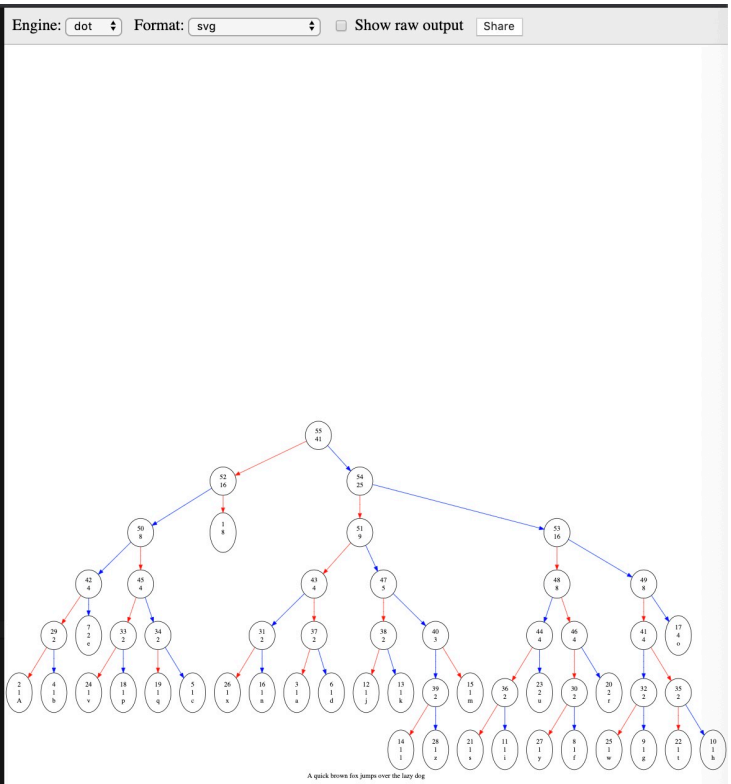
Engine: dot Format: svg Sho



```
============ A quick brown fox jumps over the lazy dog +++++++++++++++
' ' occurs  8 times
'A' occurs  1 times
'a' occurs  1 times
'b' occurs  1 times
'c' occurs  1 times
'd' occurs  1 times
'e' occurs  2 times
'f' occurs  1 times
'g' occurs  1 times
'h' occurs  1 times
'i' occurs  1 times
'j' occurs  1 times
'k' occurs  1 times
'l' occurs  1 times
'm' occurs  1 times
'n' occurs  1 times
'o' occurs  4 times
'p' occurs  1 times
'q' occurs  1 times
'r' occurs  2 times
's' occurs  1 times
't' occurs  1 times
'u' occurs  2 times
'v' occurs  1 times
'w' occurs  1 times
'x' occurs  1 times
'y' occurs  1 times
'z' occurs  1 times
===Tree built in this order===============
Leaf    node  1: Character is ' ' Weight is  8
Leaf    node  2: Character is 'A' Weight is  1
Leaf    node  3: Character is 'a' Weight is  1
Leaf    node  4: Character is 'b' Weight is  1
Leaf    node  5: Character is 'c' Weight is  1
Leaf    node  6: Character is 'd' Weight is  1
Leaf    node  7: Character is 'e' Weight is  2
Leaf    node  8: Character is 'f' Weight is  1
Leaf    node  9: Character is 'g' Weight is  1
Leaf    node 10: Character is 'h' Weight is  1
Leaf    node 11: Character is 'i' Weight is  1
Leaf    node 12: Character is 'j' Weight is  1
Leaf    node 13: Character is 'k' Weight is  1
Leaf    node 14: Character is 'l' Weight is  1
Leaf    node 15: Character is 'm' Weight is  1
Leaf    node 16: Character is 'n' Weight is  1
Leaf    node 17: Character is 'o' Weight is  4
Leaf    node 18: Character is 'p' Weight is  1
Leaf    node 19: Character is 'q' Weight is  1
Leaf    node 20: Character is 'r' Weight is  2
Leaf    node 21: Character is 's' Weight is  1
Leaf    node 22: Character is 't' Weight is  1
Leaf    node 23: Character is 'u' Weight is  2
Leaf    node 24: Character is 'v' Weight is  1
Leaf    node 25: Character is 'w' Weight is  1
Leaf    node 26: Character is 'x' Weight is  1
Leaf    node 27: Character is 'y' Weight is  1
Leaf    node 28: Character is 'z' Weight is  1
Transit node 29: Left A( 1) Right b( 1) Weight =  2
Transit node 30: Left y( 1) Right f( 1) Weight =  2
Transit node 31: Left x( 1) Right n( 1) Weight =  2
Transit node 32: Left w( 1) Right g( 1) Weight =  2
Transit node 33: Left v( 1) Right p( 1) Weight =  2
Transit node 34: Left q( 1) Right c( 1) Weight =  2
Transit node 35: Left t( 1) Right h( 1) Weight =  2
```

```
Transit node 36: Left s( 1) Right i( 1) Weight =  2
Transit node 37: Left a( 1) Right d( 1) Weight =  2
Transit node 38: Left j( 1) Right k( 1) Weight =  2
Transit node 39: Left l( 1) Right z( 1) Weight =  2
Transit node 40: Left m( 1) Right  ( 2) Weight =  3
Transit node 41: Left  ( 2) Right  ( 2) Weight =  4
Transit node 42: Left  ( 2) Right e( 2) Weight =  4
Transit node 43: Left  ( 2) Right  ( 2) Weight =  4
Transit node 44: Left  ( 2) Right u( 2) Weight =  4
Transit node 45: Left  ( 2) Right  ( 2) Weight =  4
Transit node 46: Left  ( 2) Right r( 2) Weight =  4
Transit node 47: Left  ( 2) Right  ( 3) Weight =  5
Transit node 48: Left  ( 4) Right  ( 4) Weight =  8
Transit node 49: Left  ( 4) Right o( 4) Weight =  8
Transit node 50: Left  ( 4) Right  ( 4) Weight =  8
Transit node 51: Left  ( 4) Right  ( 5) Weight =  9
Transit node 52: Left  ( 8) Right  ( 8) Weight = 16
Transit node 53: Left  ( 8) Right  ( 8) Weight = 16
Transit node 54: Left  ( 9) Right  (16) Weight = 25
Transit node 55: Left  (16) Right  (25) Weight = 41
You can see dot file at ./output/4.dot
==== Tree has 55 nodes =========
Original string cost = 287.0
Decoded  string cost = 185.0
% reduction = 35.54006968641115
```

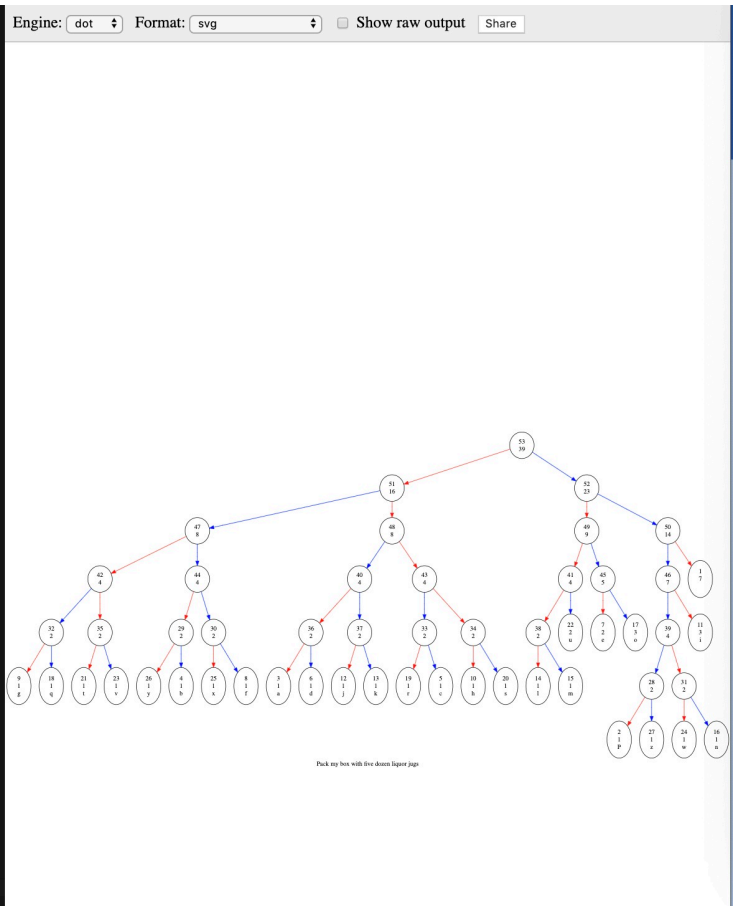
```
digraph g {
  label = " A quick brown fox jumps over the lazy dog "
  "29\n2" -> "2\n1\nA" [color=red]
  "29\n2" -> "4\n1\nb" [color=blue]
  "30\n2" -> "27\n1\ny" [color=red]
  "30\n2" -> "8\n1\nf" [color=blue]
  "31\n2" -> "26\n1\nx" [color=red]
  "31\n2" -> "16\n1\nn" [color=blue]
  "32\n2" -> "25\n1\nw" [color=red]
  "32\n2" -> "9\n1\ng" [color=blue]
  "33\n2" -> "24\n1\nv" [color=red]
  "33\n2" -> "18\n1\np" [color=blue]
  "34\n2" -> "19\n1\nq" [color=red]
  "34\n2" -> "5\n1\nc" [color=blue]
  "35\n2" -> "22\n1\nt" [color=red]
  "35\n2" -> "10\n1\nh" [color=blue]
  "36\n2" -> "21\n1\ns" [color=red]
  "36\n2" -> "11\n1\ni" [color=blue]
  "37\n2" -> "3\n1\na" [color=red]
  "37\n2" -> "6\n1\nd" [color=blue]
  "38\n2" -> "12\n1\nj" [color=red]
  "38\n2" -> "13\n1\nk" [color=blue]
  "39\n2" -> "14\n1\nl" [color=red]
  "39\n2" -> "28\n1\nz" [color=blue]
  "40\n3" -> "15\n1\nm" [color=red]
  "40\n3" -> "39\n2" [color=blue]
  "41\n4" -> "35\n2" [color=red]
  "41\n4" -> "32\n2" [color=blue]
  "42\n4" -> "29\n2" [color=red]
  "42\n4" -> "7\n2\ne" [color=blue]
  "43\n4" -> "37\n2" [color=red]
  "43\n4" -> "31\n2" [color=blue]
  "44\n4" -> "36\n2" [color=red]
  "44\n4" -> "23\n2\nu" [color=blue]
  "45\n4" -> "33\n2" [color=red]
  "45\n4" -> "34\n2" [color=blue]
  "46\n4" -> "30\n2" [color=red]
  "46\n4" -> "20\n2\nr" [color=blue]
  "47\n5" -> "38\n2" [color=red]
  "47\n5" -> "40\n3" [color=blue]
  "48\n8" -> "44\n4" [color=red]
  "48\n8" -> "46\n4" [color=blue]
  "49\n8" -> "41\n4" [color=red]
  "49\n8" -> "17\n4\no" [color=blue]
  "50\n8" -> "45\n4" [color=red]
  "50\n8" -> "42\n4" [color=blue]
  "51\n9" -> "43\n4" [color=red]
  "51\n9" -> "47\n5" [color=blue]
  "52\n16" -> "1\n8\n " [color=red]
  "52\n16" -> "50\n8" [color=blue]
  "53\n16" -> "48\n8" [color=red]
  "53\n16" -> "49\n8" [color=blue]
  "54\n25" -> "51\n9" [color=red]
  "54\n25" -> "53\n16" [color=blue]
  "55\n41" -> "52\n16" [color=red]
  "55\n41" -> "54\n25" [color=blue]
}
```



```
============ Pack my box with five dozen liquor jugs +++++++++++++++
' ' occurs  7 times
'P' occurs  1 times
'a' occurs  1 times
'b' occurs  1 times
'c' occurs  1 times
'd' occurs  1 times
'e' occurs  2 times
'f' occurs  1 times
'g' occurs  1 times
'h' occurs  1 times
'i' occurs  3 times
'j' occurs  1 times
'k' occurs  1 times
'l' occurs  1 times
'm' occurs  1 times
'n' occurs  1 times
'o' occurs  3 times
'q' occurs  1 times
'r' occurs  1 times
's' occurs  1 times
't' occurs  1 times
'u' occurs  2 times
'v' occurs  1 times
'w' occurs  1 times
'x' occurs  1 times
'y' occurs  1 times
'z' occurs  1 times
===Tree built in this order===============
Leaf    node  1: Character is ' ' Weight is  7
Leaf    node  2: Character is 'P' Weight is  1
Leaf    node  3: Character is 'a' Weight is  1
Leaf    node  4: Character is 'b' Weight is  1
Leaf    node  5: Character is 'c' Weight is  1
Leaf    node  6: Character is 'd' Weight is  1
Leaf    node  7: Character is 'e' Weight is  2
Leaf    node  8: Character is 'f' Weight is  1
Leaf    node  9: Character is 'g' Weight is  1
Leaf    node 10: Character is 'h' Weight is  1
Leaf    node 11: Character is 'i' Weight is  3
Leaf    node 12: Character is 'j' Weight is  1
Leaf    node 13: Character is 'k' Weight is  1
Leaf    node 14: Character is 'l' Weight is  1
Leaf    node 15: Character is 'm' Weight is  1
Leaf    node 16: Character is 'n' Weight is  1
Leaf    node 17: Character is 'o' Weight is  3
Leaf    node 18: Character is 'q' Weight is  1
Leaf    node 19: Character is 'r' Weight is  1
Leaf    node 20: Character is 's' Weight is  1
Leaf    node 21: Character is 't' Weight is  1
Leaf    node 22: Character is 'u' Weight is  2
Leaf    node 23: Character is 'v' Weight is  1
Leaf    node 24: Character is 'w' Weight is  1
Leaf    node 25: Character is 'x' Weight is  1
Leaf    node 26: Character is 'y' Weight is  1
Leaf    node 27: Character is 'z' Weight is  1
Transit node 28: Left P( 1) Right z( 1) Weight =  2
Transit node 29: Left y( 1) Right b( 1) Weight =  2
Transit node 30: Left x( 1) Right f( 1) Weight =  2
Transit node 31: Left w( 1) Right n( 1) Weight =  2
Transit node 32: Left g( 1) Right q( 1) Weight =  2
Transit node 33: Left r( 1) Right c( 1) Weight =  2
Transit node 34: Left h( 1) Right s( 1) Weight =  2
Transit node 35: Left t( 1) Right v( 1) Weight =  2
Transit node 36: Left a( 1) Right d( 1) Weight =  2
```

```
Transit node 37: Left j( 1) Right k( 1) Weight =  2
Transit node 38: Left l( 1) Right m( 1) Weight =  2
Transit node 39: Left  ( 2) Right  ( 2) Weight =  4
Transit node 40: Left  ( 2) Right  ( 2) Weight =  4
Transit node 41: Left  ( 2) Right u( 2) Weight =  4
Transit node 42: Left  ( 2) Right  ( 2) Weight =  4
Transit node 43: Left  ( 2) Right  ( 2) Weight =  4
Transit node 44: Left  ( 2) Right  ( 2) Weight =  4
Transit node 45: Left e( 2) Right o( 3) Weight =  5
Transit node 46: Left i( 3) Right  ( 4) Weight =  7
Transit node 47: Left  ( 4) Right  ( 4) Weight =  8
Transit node 48: Left  ( 4) Right  ( 4) Weight =  8
Transit node 49: Left  ( 4) Right  ( 5) Weight =  9
Transit node 50: Left  ( 7) Right  ( 7) Weight = 14
Transit node 51: Left  ( 8) Right  ( 8) Weight = 16
Transit node 52: Left  ( 9) Right  (14) Weight = 23
Transit node 53: Left  (16) Right  (23) Weight = 39
You can see dot file at ./output/5.dot
==== Tree has 53 nodes =========
Original string cost = 273.0
Decoded  string cost = 175.0
% reduction = 35.8974358974359
```

```
digraph g {
  label = " Pack my box with five dozen liquor jugs "
  "28\n2" -> "2\n1\nP" [color=red]
  "28\n2" -> "27\n1\nz" [color=blue]
  "29\n2" -> "26\n1\ny" [color=red]
  "29\n2" -> "4\n1\nb" [color=blue]
  "30\n2" -> "25\n1\nx" [color=red]
  "30\n2" -> "8\n1\nf" [color=blue]
  "31\n2" -> "24\n1\nw" [color=red]
  "31\n2" -> "16\n1\nn" [color=blue]
  "32\n2" -> "9\n1\ng" [color=red]
  "32\n2" -> "18\n1\nq" [color=blue]
  "33\n2" -> "19\n1\nr" [color=red]
  "33\n2" -> "5\n1\nc" [color=blue]
  "34\n2" -> "10\n1\nh" [color=red]
  "34\n2" -> "20\n1\ns" [color=blue]
  "35\n2" -> "21\n1\nt" [color=red]
  "35\n2" -> "23\n1\nv" [color=blue]
  "36\n2" -> "3\n1\na" [color=red]
  "36\n2" -> "6\n1\nd" [color=blue]
  "37\n2" -> "12\n1\nj" [color=red]
  "37\n2" -> "13\n1\nk" [color=blue]
  "38\n2" -> "14\n1\nl" [color=red]
  "38\n2" -> "15\n1\nm" [color=blue]
  "39\n4" -> "31\n2" [color=red]
  "39\n4" -> "28\n2" [color=blue]
  "40\n4" -> "36\n2" [color=red]
  "40\n4" -> "37\n2" [color=blue]
  "41\n4" -> "38\n2" [color=red]
  "41\n4" -> "22\n2\nu" [color=blue]
  "42\n4" -> "35\n2" [color=red]
  "42\n4" -> "32\n2" [color=blue]
  "43\n4" -> "34\n2" [color=red]
  "43\n4" -> "33\n2" [color=blue]
  "44\n4" -> "29\n2" [color=red]
  "44\n4" -> "30\n2" [color=blue]
  "45\n5" -> "7\n2\ne" [color=red]
  "45\n5" -> "17\n3\no" [color=blue]
  "46\n7" -> "11\n3\ni" [color=red]
  "46\n7" -> "39\n4" [color=blue]
  "47\n8" -> "42\n4" [color=red]
  "47\n8" -> "44\n4" [color=blue]
  "48\n8" -> "43\n4" [color=red]
  "48\n8" -> "40\n4" [color=blue]
  "49\n9" -> "41\n4" [color=red]
  "49\n9" -> "45\n5" [color=blue]
  "50\n14" -> "1\n7\n " [color=red]
  "50\n14" -> "46\n7" [color=blue]
  "51\n16" -> "48\n8" [color=red]
  "51\n16" -> "47\n8" [color=blue]
  "52\n23" -> "49\n9" [color=red]
  "52\n23" -> "50\n14" [color=blue]
  "53\n39" -> "51\n16" [color=red]
  "53\n39" -> "52\n23" [color=blue]
}
```

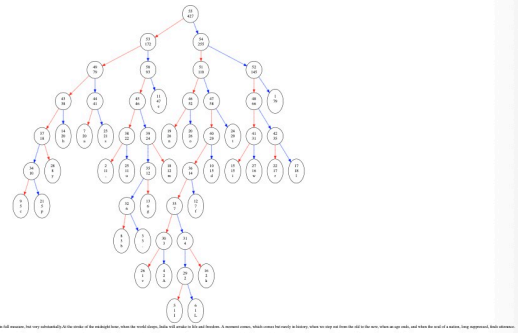


```
============ Long years ago we made a tryst with destiny, and now the time comes when we shall
redeem our pledge, not wholly or in full measure, but very substantially.At the stroke of the
midnight hour, when the world sleeps, India will awake to life and freedom. A moment comes, which
comes but rarely in history, when we step out from the old to the new, when an age ends, and when
the soul of a nation, long suppressed, finds utterance. +++++++++++++++
' ' occurs 79 times
',' occurs 11 times
'.' occurs  3 times
'A' occurs  2 times
'I' occurs  1 times
'L' occurs  1 times
'a' occurs 20 times
'b' occurs  3 times
'c' occurs  5 times
'd' occurs 15 times
'e' occurs 47 times
'f' occurs  7 times
'g' occurs  6 times
'h' occurs 20 times
'i' occurs 15 times
'k' occurs  2 times
'l' occurs 18 times
'm' occurs 12 times
'n' occurs 26 times
'o' occurs 26 times
'p' occurs  5 times
'r' occurs 17 times
's' occurs 21 times
't' occurs 29 times
'u' occurs 11 times
'v' occurs  1 times
'w' occurs 16 times
'y' occurs  8 times
===Tree built in this order==============
Leaf    node  1: Character is ' ' Weight is 79
Leaf    node  2: Character is ',' Weight is 11
Leaf    node  3: Character is '.' Weight is  3
Leaf    node  4: Character is 'A' Weight is  2
Leaf    node  5: Character is 'I' Weight is  1
Leaf    node  6: Character is 'L' Weight is  1
Leaf    node  7: Character is 'a' Weight is 20
Leaf    node  8: Character is 'b' Weight is  3
Leaf    node  9: Character is 'c' Weight is  5
Leaf    node 10: Character is 'd' Weight is 15
Leaf    node 11: Character is 'e' Weight is 47
Leaf    node 12: Character is 'f' Weight is  7
Leaf    node 13: Character is 'g' Weight is  6
Leaf    node 14: Character is 'h' Weight is 20
Leaf    node 15: Character is 'i' Weight is 15
Leaf    node 16: Character is 'k' Weight is  2
Leaf    node 17: Character is 'l' Weight is 18
Leaf    node 18: Character is 'm' Weight is 12
Leaf    node 19: Character is 'n' Weight is 26
Leaf    node 20: Character is 'o' Weight is 26
Leaf    node 21: Character is 'p' Weight is  5
Leaf    node 22: Character is 'r' Weight is 17
Leaf    node 23: Character is 's' Weight is 21
Leaf    node 24: Character is 't' Weight is 29
Leaf    node 25: Character is 'u' Weight is 11
Leaf    node 26: Character is 'v' Weight is  1
Leaf    node 27: Character is 'w' Weight is 16
Leaf    node 28: Character is 'y' Weight is  8
Transit node 29: Left I( 1) Right L( 1) Weight =  2
Transit node 30: Left v( 1) Right A( 2) Weight =  3
Transit node 31: Left k( 2) Right  ( 2) Weight =  4
```

```
Transit node 32: Left b( 3) Right .( 3) Weight =  6
Transit node 33: Left  ( 3) Right  ( 4) Weight =  7
Transit node 34: Left c( 5) Right p( 5) Weight = 10
Transit node 35: Left g( 6) Right  ( 6) Weight = 12
Transit node 36: Left  ( 7) Right f( 7) Weight = 14
Transit node 37: Left y( 8) Right  (10) Weight = 18
Transit node 38: Left ,(11) Right u(11) Weight = 22
Transit node 39: Left m(12) Right  (12) Weight = 24
Transit node 40: Left  (14) Right d(15) Weight = 29
Transit node 41: Left i(15) Right w(16) Weight = 31
Transit node 42: Left r(17) Right l(18) Weight = 35
Transit node 43: Left  (18) Right h(20) Weight = 38
Transit node 44: Left a(20) Right s(21) Weight = 41
Transit node 45: Left  (22) Right  (24) Weight = 46
Transit node 46: Left n(26) Right o(26) Weight = 52
Transit node 47: Left  (29) Right t(29) Weight = 58
Transit node 48: Left  (31) Right  (35) Weight = 66
Transit node 49: Left  (38) Right  (41) Weight = 79
Transit node 50: Left  (46) Right e(47) Weight = 93
Transit node 51: Left  (52) Right  (58) Weight = 110
Transit node 52: Left  (66) Right  (79) Weight = 145
Transit node 53: Left  (79) Right  (93) Weight = 172
Transit node 54: Left  (110) Right  (145) Weight = 255
Transit node 55: Left  (172) Right  (255) Weight = 427
You can see dot file at ./output/6.dot
==== Tree has 55 nodes =========
Original string cost = 2989.0
Decoded  string cost = 1799.0
% reduction = 39.812646370023415
```

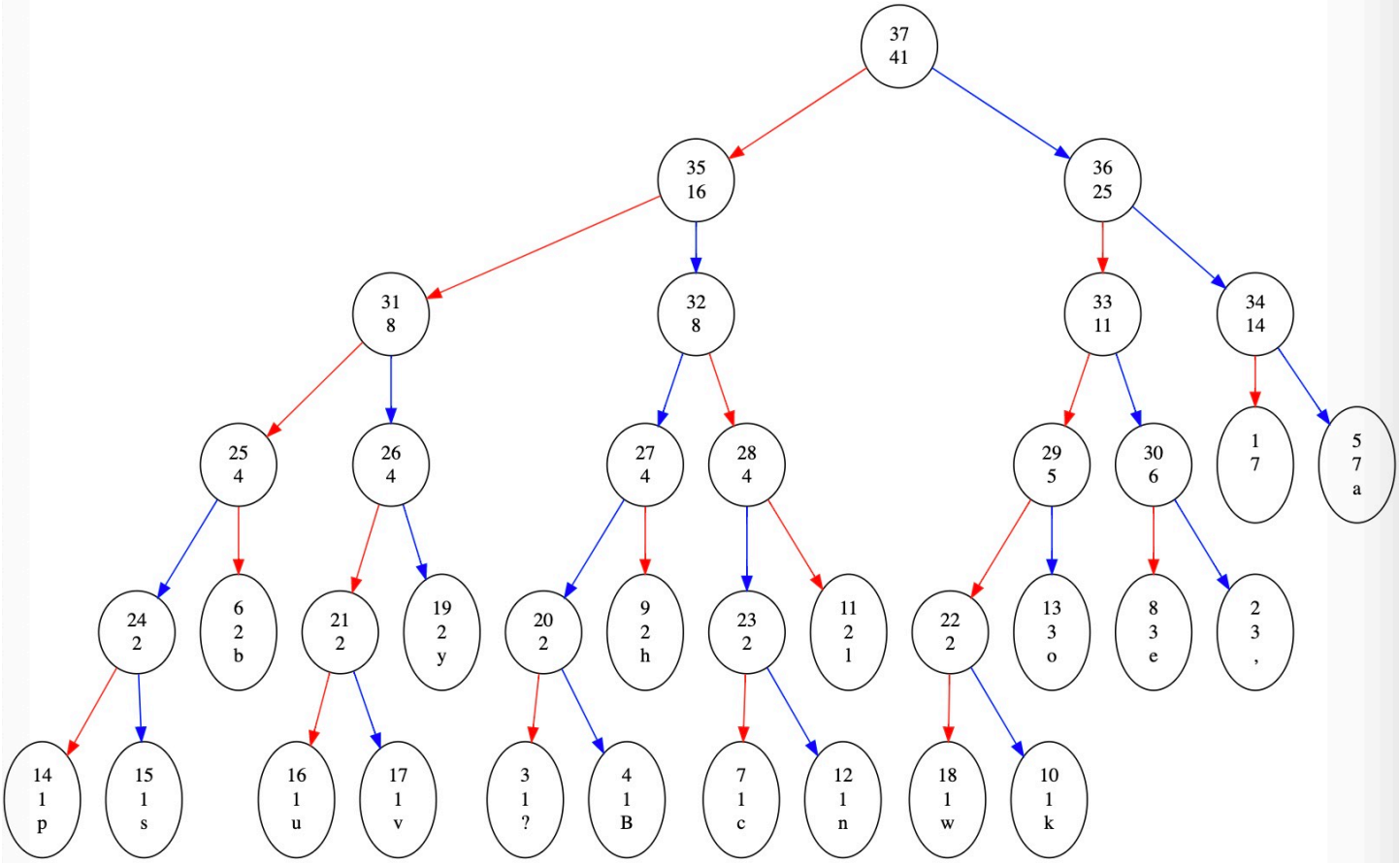
```
digraph g {
  label = " Long years ago we made a tryst with destiny, and now the time comes when we shall
  "29\n2" -> "5\n1\nI" [color=red]
  "29\n2" -> "6\n1\nL" [color=blue]
  "30\n3" -> "26\n1\nv" [color=red]
  "30\n3" -> "4\n2\nA" [color=blue]
  "31\n4" -> "16\n2\nk" [color=red]
  "31\n4" -> "29\n2" [color=blue]
  "32\n6" -> "8\n3\nb" [color=red]
  "32\n6" -> "3\n3\n." [color=blue]
  "33\n7" -> "30\n3" [color=red]
  "33\n7" -> "31\n4" [color=blue]
  "34\n10" -> "9\n5\nc" [color=red]
  "34\n10" -> "21\n5\np" [color=blue]
  "35\n12" -> "13\n6\ng" [color=red]
  "35\n12" -> "32\n6" [color=blue]
  "36\n14" -> "33\n7" [color=red]
  "36\n14" -> "12\n7\nf" [color=blue]
  "37\n18" -> "28\n8\ny" [color=red]
  "37\n18" -> "34\n10" [color=blue]
  "38\n22" -> "2\n11\n," [color=red]
  "38\n22" -> "25\n11\nu" [color=blue]
  "39\n24" -> "18\n12\nm" [color=red]
  "39\n24" -> "35\n12" [color=blue]
  "40\n29" -> "36\n14" [color=red]
  "40\n29" -> "10\n15\nd" [color=blue]
  "41\n31" -> "15\n15\ni" [color=red]
  "41\n31" -> "27\n16\nw" [color=blue]
  "42\n35" -> "22\n17\nr" [color=red]
  "42\n35" -> "17\n18\nl" [color=blue]
  "43\n38" -> "37\n18" [color=red]
  "43\n38" -> "14\n20\nh" [color=blue]
  "44\n41" -> "7\n20\na" [color=red]
  "44\n41" -> "23\n21\ns" [color=blue]
  "45\n46" -> "38\n22" [color=red]
  "45\n46" -> "39\n24" [color=blue]
  "46\n52" -> "19\n26\nn" [color=red]
  "46\n52" -> "20\n26\no" [color=blue]
  "47\n58" -> "40\n29" [color=red]
  "47\n58" -> "24\n29\nt" [color=blue]
  "48\n66" -> "41\n31" [color=red]
  "48\n66" -> "42\n35" [color=blue]
  "49\n79" -> "43\n38" [color=red]
  "49\n79" -> "44\n41" [color=blue]
  "50\n93" -> "45\n46" [color=red]
  "50\n93" -> "11\n47\ne" [color=blue]
  "51\n110" -> "46\n52" [color=red]
  "51\n110" -> "47\n58" [color=blue]
  "52\n145" -> "48\n66" [color=red]
  "52\n145" -> "1\n79\n " [color=blue]
  "53\n172" -> "49\n79" [color=red]
  "53\n172" -> "50\n93" [color=blue]
  "54\n255" -> "51\n110" [color=red]
  "54\n255" -> "52\n145" [color=blue]
  "55\n427" -> "53\n172" [color=red]
  "55\n427" -> "54\n255" [color=blue]
}
```

Engine: dot Format: svg Show raw output Share



```
============ Baa, baa, black sheep, have you any wool? +++++++++++++++
' ' occurs  7 times
',' occurs  3 times
'?' occurs  1 times
'B' occurs  1 times
'a' occurs  7 times
'b' occurs  2 times
'c' occurs  1 times
'e' occurs  3 times
'h' occurs  2 times
'k' occurs  1 times
'l' occurs  2 times
'n' occurs  1 times
'o' occurs  3 times
'p' occurs  1 times
's' occurs  1 times
'u' occurs  1 times
'v' occurs  1 times
'w' occurs  1 times
'y' occurs  2 times
===Tree built in this order===============
Leaf    node  1: Character is ' ' Weight is  7
Leaf    node  2: Character is ',' Weight is  3
Leaf    node  3: Character is '?' Weight is  1
Leaf    node  4: Character is 'B' Weight is  1
Leaf    node  5: Character is 'a' Weight is  7
Leaf    node  6: Character is 'b' Weight is  2
Leaf    node  7: Character is 'c' Weight is  1
Leaf    node  8: Character is 'e' Weight is  3
Leaf    node  9: Character is 'h' Weight is  2
Leaf    node 10: Character is 'k' Weight is  1
Leaf    node 11: Character is 'l' Weight is  2
Leaf    node 12: Character is 'n' Weight is  1
Leaf    node 13: Character is 'o' Weight is  3
Leaf    node 14: Character is 'p' Weight is  1
Leaf    node 15: Character is 's' Weight is  1
Leaf    node 16: Character is 'u' Weight is  1
Leaf    node 17: Character is 'v' Weight is  1
Leaf    node 18: Character is 'w' Weight is  1
Leaf    node 19: Character is 'y' Weight is  2
Transit node 20: Left ?( 1) Right B( 1) Weight =  2
Transit node 21: Left u( 1) Right v( 1) Weight =  2
Transit node 22: Left w( 1) Right k( 1) Weight =  2
Transit node 23: Left c( 1) Right n( 1) Weight =  2
Transit node 24: Left p( 1) Right s( 1) Weight =  2
Transit node 25: Left b( 2) Right  ( 2) Weight =  4
Transit node 26: Left  ( 2) Right y( 2) Weight =  4
Transit node 27: Left h( 2) Right  ( 2) Weight =  4
Transit node 28: Left l( 2) Right  ( 2) Weight =  4
Transit node 29: Left  ( 2) Right o( 3) Weight =  5
Transit node 30: Left e( 3) Right ,( 3) Weight =  6
Transit node 31: Left  ( 4) Right  ( 4) Weight =  8
Transit node 32: Left  ( 4) Right  ( 4) Weight =  8
Transit node 33: Left  ( 5) Right  ( 6) Weight = 11
Transit node 34: Left  ( 7) Right a( 7) Weight = 14
Transit node 35: Left  ( 8) Right  ( 8) Weight = 16
Transit node 36: Left  (11) Right  (14) Weight = 25
Transit node 37: Left  (16) Right  (25) Weight = 41
You can see dot file at ./output/7.dot
==== Tree has 37 nodes =========
Original string cost = 287.0
Decoded  string cost = 160.0
% reduction = 44.25087108013937
===============  Done with Test1 ==================
 All Hauffman Test passed. You are great. You should get an award
```

```
digraph g {
  label = " Baa, baa, black sheep, have you any wool? "
  "20\n2" -> "3\n1\n?" [color=red]
  "20\n2" -> "4\n1\nB" [color=blue]
  "21\n2" -> "16\n1\nu" [color=red]
  "21\n2" -> "17\n1\nv" [color=blue]
  "22\n2" -> "18\n1\nw" [color=red]
  "22\n2" -> "10\n1\nk" [color=blue]
  "23\n2" -> "7\n1\nc" [color=red]
  "23\n2" -> "12\n1\nn" [color=blue]
  "24\n2" -> "14\n1\np" [color=red]
  "24\n2" -> "15\n1\ns" [color=blue]
  "25\n4" -> "6\n2\nb" [color=red]
  "25\n4" -> "24\n2" [color=blue]
  "26\n4" -> "21\n2" [color=red]
  "26\n4" -> "19\n2\ny" [color=blue]
  "27\n4" -> "9\n2\nh" [color=red]
  "27\n4" -> "20\n2" [color=blue]
  "28\n4" -> "11\n2\nl" [color=red]
  "28\n4" -> "23\n2" [color=blue]
  "29\n5" -> "22\n2" [color=red]
  "29\n5" -> "13\n3\no" [color=blue]
  "30\n6" -> "8\n3\ne" [color=red]
  "30\n6" -> "2\n3\n," [color=blue]
  "31\n8" -> "25\n4" [color=red]
  "31\n8" -> "26\n4" [color=blue]
  "32\n8" -> "28\n4" [color=red]
  "32\n8" -> "27\n4" [color=blue]
  "33\n11" -> "29\n5" [color=red]
  "33\n11" -> "30\n6" [color=blue]
  "34\n14" -> "1\n7\n " [color=red]
  "34\n14" -> "5\n7\na" [color=blue]
  "35\n16" -> "31\n8" [color=red]
  "35\n16" -> "32\n8" [color=blue]
  "36\n25" -> "33\n11" [color=red]
  "36\n25" -> "34\n14" [color=blue]
  "37\n41" -> "35\n16" [color=red]
  "37\n41" -> "36\n25" [color=blue]
}
```



Baa, baa, black sheep, have you any wool?