```
CL-USER> (defconstant message
  '(
              no yes not yes yes not no no no is is hello yes no no yes not no
              not no hello is goodbye no yes when no yes not no not no
              no yes not is why why why yes is how is yes is why no yes not
              no no is is not no yes no yes hello yes hello yes no no yes
              no hello goodbye no is goodbye no yes when
    ))
MESSAGE
CL-USER> (defconstant tree (make-huffman-tree message))
TREE
CL-USER> tree
(((HOW WHEN GOODBYE WHY HELLO YES NOT IS NO) 74)
 (((HOW WHEN GOODBYE WHY HELLO YES) 32)
  (((HOW WHEN GOODBYE WHY HELLO) 15)
   (((HOW WHEN GOODBYE) 6) (((HOW WHEN) 3) (((HOW) 1)) (((WHEN) 2)))
    (((GOODBYE) 3)))
   (((WHY HELLO) 9) (((WHY) 4)) (((HELLO) 5))))
  (((YES) 17)))
 (((NOT IS NO) 42) (((NOT IS) 19) (((NOT) 9)) (((IS) 10))) (((NO) 23))))
CL-USER> (defconstant binary (encode message tree))
BINARY
CL-USER> binary
(1 1 0 1 1 0 0 0 1 0 1 1 0 0 1 1 1 1 1 1 1 0 1 1 0 1 0 0 1 1 0 1 1 1 1 1 0 1 1
 0 0 1 1 1 0 0 1 1 0 0 1 1 1 0 1 0 0 0 1 1 1 0 1 0 0 0 0 1 1 1 0 1 1 0 0 1 1 1
 0 0 1 1 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 1 1 0 1 0 0 0 0 0 1 0 1
 0 1 1 0 1 0 0 1 0 1 1 0 1 1 0 0 1 1 1 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 1 0 1 0 0
 1 1 0 1 0 0 1 1 0 1 1 1 1 1 0 1 1 1 0 0 1 1 0 0 0 1 1 1 1 0 1 0 0 0 1 1 1 0 1
 0 0 0 0 1)
CL-USER> (decode binary tree)
(NO YES NOT YES YES NOT NO NO NO IS IS HELLO YES NO NO YES NOT NO NOT NO HELLO
 IS GOODBYE NO YES WHEN NO YES NOT NO NOT NO NO YES NOT IS WHY WHY WHY YES IS
 HOW IS YES IS WHY NO YES NOT NO NO IS IS NOT NO YES NO YES HELLO YES HELLO YES
 NO NO YES NO HELLO GOODBYE NO IS GOODBYE NO YES WHEN)
CL-USER>
```