* opening\_text():

Only constant operations are performed.

* O(1)
* choose\_diff():

Due to the while loop.

* O(n)
* draw\_gallows(stage=0, life=0, hints=[]):

Only constant operations are performed

* O(1)
* alphabets(letter="", abc=[], abc\_used=[], print\_check=False):

Due to the 3 independent for loops

And .remove method is O(n)

* O(n + n + n + n) = O(n)
* word\_blanks(word):

Due to the 3 independent for loops

* O(n + n) = O(n)
* letter\_check(letter, word, blanks):

Due to the for loop

* O(n)
* guess\_validity(input\_letter, abc\_used, hint\_list):

Due to the while loop.

* O(n)
* win\_check(blanks):

Due to the for loop

* O(n)
* get\_random\_word():

Constant operations are performed and choose\_diff() is called

* O(1 + n) = O(n)
* hints(hint\_num=0, letters=[], blanks=[], avail\_letters=[], used\_letters=[]):

Due to the while loop.

* O(n)
* hangman(word):

functions called,

* draw\_gallows() = O(1)
* alphabets() = O(n)
* guess\_validity() = O(n)
* hints() = O(n)
* letter\_check() = O(n)
* win\_check() = O(n)
* .remove() = O(n)

O(1+n+n+n+n+n+n) = O(n)

Worse case scenario the while loop runs, O( 10 + n (length of word) ), times. So, O(n)

Therefor overall time of hangman() is,

O(n\*n) = O(n2)

* main():

get\_random\_word(), = O(n)

hangman(word), = O(n2)

O(n + n2) = O(n2)

As it is called in a while loop,

O(n2 \* n) = O( n3)