$O(n) = O(\frac{n}{2})$ or $n * 4$

dominant term

"to" = in terms of

BUT

$O(\frac{n}{2})$ is, in reality, twice as fast as $O(n)$ in clock time

Make spell checker

$\frac{7}{2}$ ~~$\frac{3}{2}$~~ (1)

Note: if (a>5)
{
result = 3
{
else
{
result = 4
{

is

result = a>5 ? 3 : 4

(1) ?: → Known as the ternary operator

---

(2) struct rusage contains ~~data types~~ variables pertaining to CPU usage

So struct rusage before, after;

simply declares to sets of this data, how much CPU is used before and after the program runs.

---

(3) getrusage is a function in sys/resource.h takes 2 parameters:

1 int who → RUSAGE_SELF

2 struct rusage *usage

Sum of resources used by function → a pointer to rusage struct

getrusage returns 0 when successful and -1 when failed

AND since rusage is a pointer

(3) continued all the variables in that structure get updated when getrusage is called

(4) we use getrusage before loading the dictionary and after loading the dictionary, presumably to [redacted] track efficacy during that process

(5) calculate(&before, &after)

* by passing pointers we can directly interface w/ the values stored at address of before.

a. if these pointers are NULL exit

b. otherwise, perform the following arithmetic and return result

* All converted to [redacted] Seconds

(After) UserCPU [redacted] (seconds) + UserCPU [redacted] (micro-)

(Before) UserCPU [redacted] (seconds) + UserCPU [redacted] (micro)

(After) KernelCPU [redacted] (seconds) + [redacted] (micro) -

(Before) [redacted]