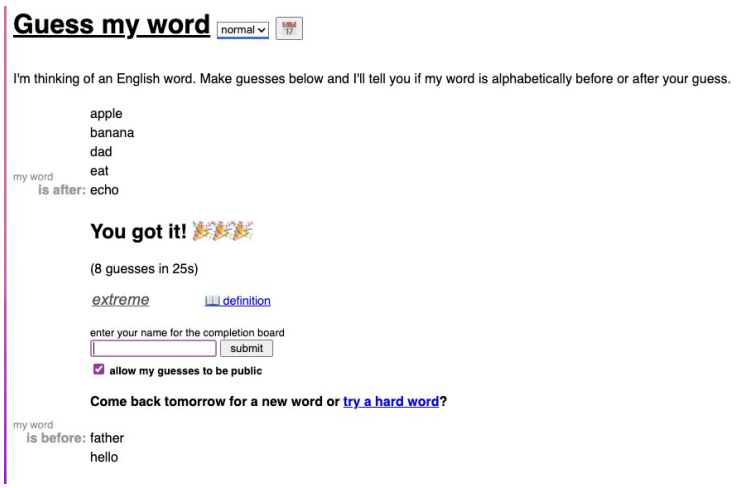
(a).

P2



(b) We can use binary search and set a pivot as a mid index, then we get the value of mid index and compare with our findword to check if this word is before or after our mid word. We can execute repeatedly until we find our word.

```
// Pseudocode using Binary search
findSercertWords(words, k){
    int left = 0, right = 2^k - 1;
    int mid = left + ((right - low) / 2);
    midword = words[mid];

    while(left <= right){
        if(wordfind < midword){
            right = mid - 1;
        }else if (wordfind > midword){
            left = mid + 1;
        }else{
            return wordfind;
        }
    }
}
```

Because we implement this question as a Binary search, so, we will guess the time is $\log(2^k - 1) = k$.

so, we can get k time guess at most.

(c).

As we know, the maximum guess number is $T(n)$, we can apply for the binary search algorithm, we will get $T(n) = T(n/2) + O(1)$

then, we applying the Master Theorem, we can know $a=1$, $b=2$, and we also have

$$n^{(\log_b^a)} = n^0 = 1.$$

so, we can get $T(n) = \theta(\log n)$.

(d). Because the average guessing time would be

$$\log_2^{(26751)} \approx 18$$

And $18 \cdot \$1 > \$15$.

so, you will lose the money.

second solution:

$$E(x) = P(\text{win}) \cdot 15 - P(\text{lose}) \cdot 1$$

$$= \frac{1}{267751} \times 15 - \frac{267750}{267751} \times 1$$

$$= -0.9999$$

so, we can get the same conclusion, lose the money.