Talk
Example
Brute Force
Optimize
Walk Through

Implementation
Test

Alyssa Lingad
MCP #4

WORD BREAK

TALK

- input: string, dict of words
- output: boolean (if string can be split into seq of dict words
- null or empty str?

$$\begin{tikzcd} \text{null} \arrow[d] & \text{empty str} \arrow[d] \\ \text{exception} & \text{false} \end{tikzcd}$$

- empty dict? FALSE
- same word can appear multiple times in ~~[illegible]~~ input str
- "Hello" and "hello" are equal
- str only alphabetic chars

EXAMPLE

	INPUT	CLASS	OUT
✓	null, {pear}	null str	exception
✓	"", {pear}	empty str	False
✓	"pear", {}	empty dict	False
✓	"HelloHello", {hello}	sameword mult. times	True
✓	"Pearfruit", {pear, fruit}	capitalized str	True
✓	"youenjoy", {you, enjoy}	basic case	True
✓	"paris", {france}	not able to segment	false

BRUTE FORCE $O(N^2)$ solution

```
canBeSegmented(str s, dict d):
i = 0, s.lowercase()
while i < length(s):
    check = False
    ~~[illegible]~~ j = i+1
    while j < length(s):
        if s[i:j] in d:
            check = True
            break                      <- j++
    if !check:
        return False
    i = j                              <- if j == length(s)-1:
                                              return True
return check
```

OPTIMIZE

- handle erroneous input
- cannot think of a way to optimize :(

WALK THROUGH

INPUT VALIDATION

↓

loop through input str

↓

loop again : check if word is in dict until end of string

↓

move index to index+1 of where last word was found

↓

return check

IMPLEMENTATION

```
                                                                set
   public static boolean canBeSegmented(string s, Hash~~Map~~ d){
      int i=0;
      [illegible]
      if(s == null)
         throw new Exception();
      if (s == "")
         return False;
      if(d.size() < 1)
         return False                                   (s.lowercase()
      while (i < s.length()){
         boolean check = False;
         int j = i+1;
         while (j < s.length()){
            if(s.substring(i,j) in d){
               check = True;
               break;
         if (!check)
            return False;
         i=j;
      return check;
```

TEST

same cases as in example