# داکیومنت پروژه دوم

## شرح الگوریتم:

به طور کلی می‌توان این الگوریتم را به **دو بخش اصلی** تقسیم کرد؛ بخش اول به ایجاد و **حدس کلمه ای** بر اساس قوانین به دست آمده از ورودی‌های کاربر می‌پردازد (در صورت وجود) و بخش دوم به **دریافت بازخورد** کاربر به کلمه‌ی حدس زده مذکور و تحلیل آن که منجر به یافتن قوانین جدید می‌شود. در اینجا این دو بخش باید حداکثر ۱۰ بار تکرار شده تا به کلمه موردنظر برسیم و در غیر این صورت کاربر شکست می‌خورد. واضح است که منطق بازی مربوط به بخش اول یعنی حدس کلمه می‌باشد.

می‌دانیم کلمه باید چهار حرف انگلیسی باشد که الزاما معنی ای ندارند. تنها قانون در ابتدای بازی آن است که حروف تکراری در کلمه مورد نظر نداریم. بنابراین می‌توان درختی ۵ سطحی ایجاد کرد که در سطح صفر، ریشه‌ی درخت، و در هر سطح ۱ تا ۴، به عنوان فرزندان هر گره در سطح i-1، تمام حروف انگلیسی به صورت گره‌های مجزا قرار دارند و این گره‌ها در سطح i بیانگر حروف قابل قرارگیری در جایگاه i ام کلمه مورد نظر است، به صورتی که حرف جایگاه قبل، پدر این گره می‌باشد. (i از ۱ تا ۴).

**نکته:** در واقع فرزندان هر گره از گره سطح بالاتر، تمام حروف انگلیسی خواهند بود، بنابراین به عنوان مثال در سطح دوم ما ۲۶ * ۲۶ گره داریم.

در هربار که برنامه کلمه جدیدی را حدس می‌زند، از ریشه شروع کرده و اولین گره مجاز را انتخاب می‌کند و با فرض قرارگیری آن در کلمه حدس زده شده، از طریق فرزندان آن گره وارد سطح بعدی می‌شود. گفته شد الگوریتم در هر سطح اولین انتخاب مجاز مربوط به آن سطح را انتخاب می‌کند و پیش می‌رود؛ اما بر اساس چه قوانینی؟ واضح است در ابتدا تنها قانون، عدم وجود حرف تکراری می‌باشد. بنابراین الگوریتم ما همواره کلمه‌ی "abcd" را به عنوان اولین حدس تولید می‌کند. در بخش بعد که کاربر بازخورد را به برنامه می‌دهد، برنامه بر اساس علامت‌های + ,= و - که معنای آنهارا می‌دانیم، یکی از این سه کار را انجام می‌دهد:

**نکته:** باید قوانین را در جایی ذخیره کنیم که این کار را به وسیله یک دیکشنری (notInPlace) انجام می‌دهیم که کلید های آن شماره سطوح از ۱ تا ۴ و مقادیر هر کلید، حروف غیر امیدبخش در آن سطح است. در واقع ما می‌خواهیم بدانیم در هر سطح، چه حروف و گره‌هایی غیرامیدبخش هستند و آن هارا ادامه ندهیم.تابع feedbackAnalisys():

**۱. اگر بازخورد "=" باشد:**

حرف حدس زده شده درست است و بنابراین باقی حروف نمی‌توانند در این جایگاه قرار گیرند. بنابراین تمام حروف الفبا به جز حرف مذکور را، به مقادیر کلید سطح مربوطه در دیکشنری اضافه می‌کنیم.

**۲. اگر بازخورد "+" باشد:**

حرف مورد نظر تنها در این جایگاه نمی‌تواند باشد و در باقی جایگاه‌ها وجود دارد. بنابراین تنها این حرف را به مقادیر کلید سطح مربوطه در دیکشنری اضافه می‌کنیم.

**نکته:** در این الگوریتم نیازی نیست که بررسی کنیم چه حروفی را مطمئن هستیم که در کلمه نهایی وجود دارند، از آنجا که حروف انگلیسی از اول به آخر پیمایش می‌شود، اگر حرفی بازخورد "+" بگیرد، هرگز جا انداخته نمی‌شود.

**۳. اگر بازخورد "-" باشد:**

حرف در هیچ کدام از ۴ جایگاه کلمه نهایی وجود ندارد. بنابراین این حرف را به مقادیر تمام کلید های دیکشنری اضافه می‌کنیم.

پس از به‌روزرسانی دیکشنری مذکور، در صورتی که بازخورد کاربر "====" نباشد، در صورت داشتن شانس مجدد، الگوریتم بار دیگر اجرا می‌شود و اینبار با قوانین جدید کلمه را می‌سازد. در واقع برای ساخت کلمه به این صورت عمل می‌کند: ( تابع backtrackingWordle() ):

حرکت در درخت انتزاعی را از ریشه شروع کرده و به برگ ختم می‌کنیم. بنابراین حروف کلمه، حرف به حرف از a به z بررسی می‌شوند. در ابتدا کلمه حدس زده شده حاوی هیچ حرفی نیست (چهار تا صفر در نظر گرفته می‌شود)، در هر سطح، الگوریتم تمام حروف انگلیسی را دانه به دانه بررسی می‌کند (از a تا z)، در دو صورت حرف غیر امید بخش است: ۱. قبلا در کلمه وجود داشته باشد (در سطوح قبل) ۲. در دیکشنری، در مقادیر کلید مربوط به آن سطح وجود داشته باشد. در غیر این صورت اولین حرف امید بخش انتخاب شده و به سراغ گره‌های فرزند آن می‌رویم.

**نکته:** بخش عقب‌گرد آن‌جایی اتفاق می‌افتد که حرفی انتخاب می‌شود و سپس به سطح بعد می‌رویم و می‌بینیم در میان گره‌های فرزند آن گره، هیچ گره امید بخشی وجود ندارد (و هنوز ۴ حرف را حدس نزدیم)، بنابراین آخرین گره‌ای که انتخاب شده بود را باید به بالا برگردیم و به سراغ گره امید بخش بعد از آن برویم.

**نکته:** از آنجا که الگوریتم یک کار یکسان را در هر سطح انجام می‌دهد، می‌توان آن را به صورت بازگشتی برای هر سطح نوشت و واضح است که شرط خروج آن است که تمام چهار حرف حدس زده شده باشند. در واقع این تابع برای هر بار حدس کلمه حداقل ۴ بار فراخوانی می‌شود که در هربار، یک حرف مورد تایید در یکی از ۴ جایگاه را مشخص می‌کند.

**نکته:** از آنجا که ورودی‌های کاربر و همچنین کلمه‌ی نهایی valid می‌باشند، مطمئن هستیم که الگوریتم حتما یک کلمه حدس را تولید می‌کند.

تمام آنچه گفته شد را می‌توان به کمک سه تابع پیاده سازی کرد که شبه کد آن به صورت زیر است:

(دو تابع مذکور توسط تابع display() به ترتیب فراخوانی می‌شوند و همچنین این تابع شانس‌های موجود و یا برد و باخت را مدیریت می‌کند).

```
CONSTANTS:
    ALPHABET = ['a', 'b', 'c', ..., 'z']

GLOBAL VARIABLES:
    notInPlaces = {0: [], 1: [], 2: [], 3: []}
    guessedWord = ['0', '0', '0', '0']

FUNCTION feedbackAnalysis(guessedWord, remainingChances):
    PRINT Guessed Word and Remaining Chances and ask for feedback

    IF feedback == '====':
        Word Guessed Successfully, return True

    FOR index FROM 0 TO 3:
        IF feedback[index] == '=':
            ADD all other letters to notInPlaces[index]
        ELSE IF feedback[index] == '+':
```

```
            Add the letter to the non promising letters
            in level index (notInPlaces[index])
        ELSE IF feedback[index] == '-':
            Add the letter to the non promising letters in all levels (notInPlaces[0..3])

    RETURN False cause word not guessed

FUNCTION backtrackingWordle(index):
    IF index == 4:
        RETURN True

    FOR letter IN ALPHABET:
        IF letter IN guessedWord OR letter IN notInPlaces[index]:
            CONTINUE

        guessedWord[index] = letter

        IF backtrackingWordle(index + 1):
        RETURN True

        guessedWord[index] = '0'

    RETURN False

FUNCTION display():
    WHILE there are remaining chances:
        guessedWord = ['0', '0', '0', '0']
        CALL backtrackingWordle(0)

        IF feedbackAnalysis(guessedWord, remainingChances):
            PRINT "Word Guessed!"
            BREAK

        decrement remaining chances
    IF chances are over:
        PRINT "Can't guess the word"

MAIN:
    CALL display()
```

نکته: در کد اصلی همراه با فانکشن های بالا، در فانکشن آنالیز فیدبک، ارورهندلینگی نیز اضافه شده است که در صورت خطای کاربر هنگام ورودی، مجددا ورودی بگیرد. همچنین اگر کاربر از نماد اشتباهی استفاده کند نیز این اتفاق می‌افتد.

مهسا حق‌نویس ـ مهدی محمودخانی