

CS 1.2: Intro to Data Structures & Algorithms

Sampling Worksheet

Name: Joseph Paul

Sampling

Assume you have a histogram data structure that stores the following words and counts:

```
histogram = [('cats', 3), ('dogs', 4), ('rabbits', 2), ('turtles', 1)]
```

Review the code below that implements a non-uniform sampling function given a histogram:

```
def sample(histogram):  
    """Return a word from this histogram, randomly sampled by weighting  
    each word's probability of being chosen by its observed frequency."""  
    tokens = sum([count for word, count in histogram]) # Count total tokens  
    dart = random.randint(1, tokens) # Throw a dart on the number line  
    # Note: Assume that randint returns 8 here and dart stores the value 8  
    fence = 0 # Border of where each word splits the number line  
    for word, count in histogram: # Loop over each word and its count  
        fence += count # Move this word's fence border to the right  
        if fence >= dart: # Check if this word's fence is past the dart  
            return word # Fence is past the dart, so choose this word
```

Q7: Execute the code above as the Python interpreter would. Complete the table below to keep track of the value of each variable inside the for loop. Write “N/A” if a value is never evaluated.

Iteration	word	count	fence	dart	fence >= dart
1	'cats'	3	3	8	False
2	'dogs'	4	7	8	False
3	'rabbits'	2	9	8	True
4	'turtles'	1	N/A	8	N/A

Q8: Which word is returned when the `sample` function is executed? (Assume `dart`'s value is 8.)

`'rabbits'`

Q9: Mark the number line below to show each word's count and fence values from the table above and where the value of `dart` is on the number line to determine which word is returned:

