Assignment 1 Review
Part 4: make_seq.py

```python
# Part 4
### TODO Generate a random nucleotide sequence

# Initialize an empty string that nucleotides can be appended to
sequence = ""

# Create a for loop that will be repeated <sequence_length> times
for i in range(sequence_length):
    # Generate a random decimal
    rand = random.random()
    # Use if/else if/else logic to determine which nucleotide to add
    # Append the nucleotide to the nucleotide sequence
    if (rand < a_freq):
        sequence += "A"
    elif (rand < a_freq + c_freq):
        sequence += "C"
    elif (rand < a_freq + c_freq + g_freq):
        sequence += "G"
    else:
        sequence += "T"

# Print the full nucleotide sequence
print(sequence)
```
Part 4: make_seq.py

- random.random() will choose a random decimal in [0,1)
- Since we are creating a sequence with 1,000,000 nucleotides, we are choosing a random decimal enough times to get values covering the whole range of [0,1)

```
a_freq | c_freq | g_freq | t_freq
0      | a_freq | a_freq + c_freq | a_freq + c_freq + g_freq
```
Part 4: make_seq.py

- By using if and elif logic, the script will check the next condition for our random decimal if the previous condition has not been met, which prevents us from adding more than one nucleotide each time we generate a new random decimal.
- For example, in our case, a_freq = .22 and c_freq = .28, so if the first random decimal is .15, we will add an ‘A’ and generate a new decimal.
- If the first random decimal is .25, the first if condition (rand < a_freq) is false, so we check the next condition, which is true, so we add a ‘C’ and generate a new random decimal.
Part 5: Counting Dinucleotides with Overlapping Window

```python
# Initialize empty dictionary for dinucleotides
dinuc_dict = {}

## Part 5
## TODO Use overlapping windows to count the dinucleotides. See the assignment for more information on overlapping windows.
# Iterate thru integers 0 to length of sequence - 1, (e.g. if length of sequence is six bases, will iterate thru 0, 1, 2, 3, 4, 5)
for i in range(len(nucleotides)):
    dinuc = nucleotides[i:i+2]  # Slice sequence string to grab bases at positions i, i+1
    # This condition checks to see if we have reached the last base in our sequence, meaning there is no more dinucleotides to check
    if (len(dinuc) == 1):
        break  # Exits the loop completely
    # This condition checks to see if our current dinucleotide contains an 'N'
    if ('N' in dinuc):
        continue  # If there is an 'N' in dinucleotide, we move on to next i value in our loop without checking the two if statements
    # This condition checks if the current dinucleotide is already a key in our dictionary
    if dinuc not in dinuc_dict.keys():
        dinuc_dict[dinuc] = 1  # If dinucleotide is NOT in dictionary keys, we set the count for that dinucleotide as 1
    else:
        dinuc_dict[dinuc] += 1
```
Part 5: Counting Dinucleotides with Overlapping Window

`str.count()` is a non-overlapping counting method.
Part 5: Counting Dinucleotides with Overlapping Window

**Question 3.1:** What are the dinucleotide frequencies from chr20?

Dinucleotide Frequencies:
- **AA:** 0.088
- **AC:** 0.05
- **AG:** 0.073
- **AT:** 0.068
- **CA:** 0.075
- **CC:** 0.058
- **CG:** 0.012
- **CT:** 0.073
- **GA:** 0.082
- **GC:** 0.048
- **GG:** 0.06
- **GT:** 0.051
- **TA:** 0.055
- **TC:** 0.061
- **TG:** 0.076
- **TT:** 0.091

**Question 3.2:** What are the dinucleotide frequencies for the generated `random_seq_1M.txt`. (Since the sequence was generated randomly, your frequencies may be slightly different.)

Dinucleotide Frequencies:
- **AA:** 0.077
- **AC:** 0.061
- **AG:** 0.061
- **AT:** 0.079
- **CA:** 0.06
- **CC:** 0.048
- **CG:** 0.048
- **CT:** 0.062
- **GA:** 0.082
- **GC:** 0.048
- **GG:** 0.049
- **GT:** 0.062
- **TA:** 0.079
- **TC:** 0.061
- **TG:** 0.063
- **TT:** 0.08*