Artificial Intelligence Term Project Spring 2017

0556562 陳鴻君

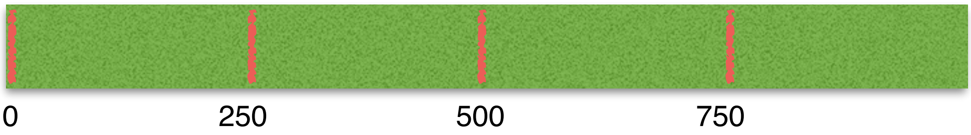
Definition:

1. If a pattern appears more than 2 times, I consider it as randomness.
2. I need to find out a significant pattern in every sequence in a family.
3. This significant pattern need to fit TA's constraint. (Goal test)
4. Fast search algorithm is used.

Method:

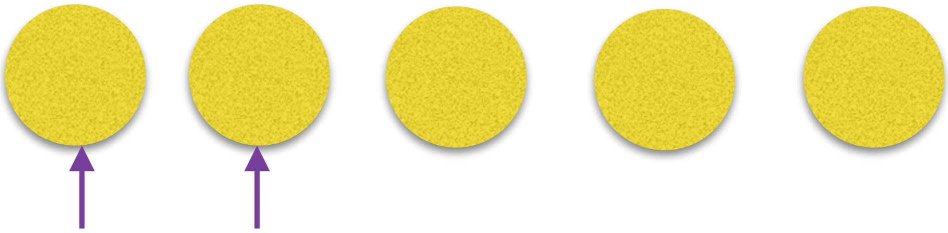
1. In Q1, Brute force is used. I search S1 in Q1 from index = 0 and length = 15. Therefore in the worst case, it cost 1000-15=985 times.
2. In Q2 and Q3, there're mutations. But significant is still appeared in each sequence. The only thing I need to do is to search pattern's mutation in a sequence (e.g., S1).

To speed up, I partition a sequence into 4 partitions as below.

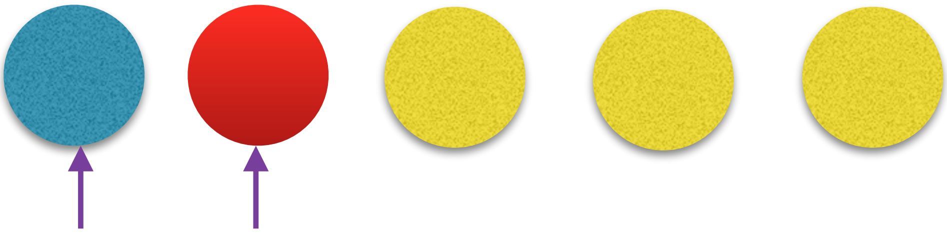


Searching each part each time, it control by a variable called "turn". For instance, first time I search index = 250 and length = 15, then next time I will search index = 500 and length = 15.
If I define 0-249 as G1, 250-499 as G2, 500-749 as G3, and 750-1000 as G4. The order is G2G3G1G4.

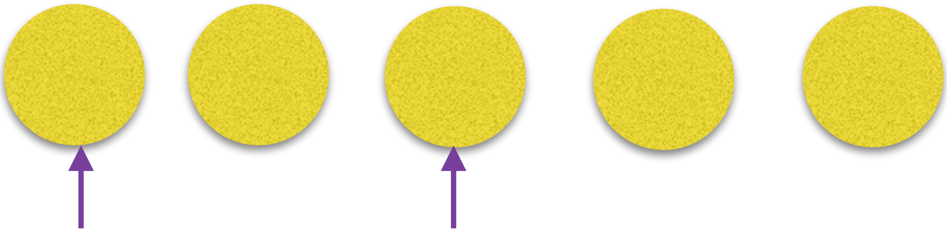
In each search, the pattern needs to be mutated. (e.g., index = 250, length =15, mutation = 5)
I use a bubble-sort-like mutation method.
For example, length = 5, mutation = 2.



Step1: Select candidate.



Step2: Each candidate mutate 4 times. So total mutate 3^2 = 9 times in this instance.



Step3: Bubble up.
Step4: repeat step2 and 3 until all members are mutate.

Result:

Q1:
Significant = ATTACCGTTAAGCTG
Time cost = 0.12 seconds

Q2:
Significant = TCAATGATATAAATA
Time cost = 58.19 seconds

Q3:
Significant = TTAGCCTAAAGACCT
Time cost = 0.11 seconds