Proof of ABRACADABRA Problem and a General Program to Solve it

马浩博 518030910428

July 2, 2020

1 Introduction

I saw this problem in E10.6 in the textbook. And I think this problem is interesting, so I'd like to write a report and also write a program to slove all the questions like this.

First let me introduce the question. Just like the famous Infinite monkey theorem, we let a monkey type a capital letter at random at each of times 1,2,3,.... The sequence of letters typed forms an IID sequence of RVs each chosen uni-formly from amongst the 26 possible capital letters.

Now let T be the first time by which the monkey has produced the consecutive sequence S =ABRACADABRA. Prove that:

$$E(T) = 26^{11} + 26^4 + 26$$

2 Proof

Proof. To solve this problem, let's first consider some gamblers. Just before each time $n = 1, 2, ...$, a new gambler arrives on the scene. And he bets \$1 that

$$\text{the } n^{th} \text{ letter will be } A.$$

If he loses, he leaves. If he wins, he receives \$26 all of which he bets on the event that

$$\text{the } (n+l)^{th} \text{ letter will be } B.$$

If he loses, he leaves. If he wins, he bets his whole current fortune of $\$26^2$ that

$$\text{the } (n+2)^{th} \text{ letter will be } R.$$

and so on through sequence S.

Because \$1 can win \$26 with a 1/26 probability or get nothing, obvously this is a fair game and the expectation of net winnings should be 0.

Assume finally the gambler N won 11 times in a row in the first time, then $T = N + 10$.

At this time, the 1 to $N-1$ gambler has lost all money and left, the N gambler won a lot and has $\$26^{11}$, $N+1$ to $N+6$ also lost all money and left, $N+7$ gambler hitted ABRA so he holds $\$26^4$, the $N+8$ and $N+9$ also lose all money, the $N+10$ gambler won the last letter A so holds \$26. And the $N+11$ gambler and subsequent gamblers have not entered the game and still hold the initial funds \$1.

The k gambler’ s asset immediately after the monkey hits the n character is recorded as $X_n^{(k)}$, which can be claimed to be a martingale because this is a fair game.So the total profit of all gamblers at time n is

$$M_n = \sum_{k=1}^{\infty} \left(X_n^{(k)} - 1 \right)$$

which is also a martingale. Before typing, we have $M_0 = 0$. According to the previous discussion we know that

$$M_T = \sum_{k=1}^{T} \left(X_n^{(k)} - 1 \right) = 26^{11} + 26^4 + 26 - T$$

By Doob's Optional-Stopping Theorem(10.10(c) in the textbook), we can get that

$$E(M_T) = E(M_0) = 0$$

Thus

$$E(T) = 26^{11} + 26^4 + 26$$

$\square$

3 Program

Then I think about the ganeral situation. Let T_S be the first time by which the monkey has produced the consecutive sequence S, how can we quikly get $E(T_S)$?

Considering our proof process, it is not difficult to tell that this is a self-matching process. So we can use the traditional Knuth–Morris–Pratt(KMP) algorithm to efficiently calculate it.

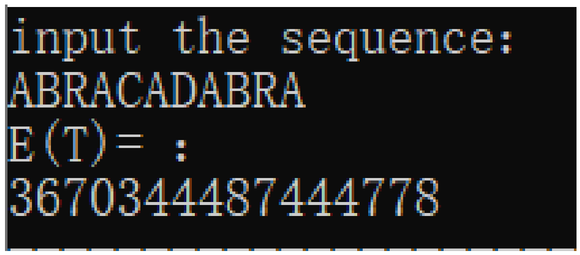
But it's not that simple. We need to calculate the length of all the possible prefix strings that equals a suffix string. And I want to calculate it within $O(n)$, which means we must use the most efficient way. After thinking, I came up with a method that uses the next array in KMP to calculate. We know that next[i] in KMP means the length of the longest common element of prefix and suffix in the first i characters(If you do not know the KMP algorithm, you can treat it as a consistent conclusion), so we can calculate it by the following code in C++:

```cpp
int l=strlen(S);
GetNext(S);
ans[0]=l;
int n=0,i=l;
while(next[i]>0)
{
 n=n+1;
 ans[n]=next[i];
 i=next[i];
}

```

The GetNext() function is to get next array in KMP. If we let S = ABRACADABRA, then we can get $\{11, 4, 1\}$ in the ans array. Obvously, this algorithm is efficient.

I finish the program in Solve.cpp, and it works fine:



Now, we have a general solution to the similar questions.