Computing Science 204 (B1)

Position where First Head Appears in n Tosses of a Fair Coin

Monday, July 17

In this note we want to analyze the algorithm below, which finds the position of the first occurrence of a head in n tosses of a fair coin.

INPUT: A sequence of n independent tosses of an unbiased coin (that is, both H and T are equally likely).

PROBLEM: Find the position where the first H appears.

ALGORITHM:

```
void first_head()
{
   int  n, position;
   char next_toss;

   scanf("%d",&n);
   scanf("%c",&next_toss);
   position = 1;
   while (position < n && next_toss == 'T')
   {
      position++;
      scanf("%c",next_toss);
   }
   if (next_toss == 'H')
      printf(("First Head at Location: %d\n",position);
   else
      printf("No Heads Found\n");
}
```

Worst Case Complexity:

Clearly, the worst case occurs when the first H is in the last position, or when there is no H at all. In this case, we need to look at all n tosses, that is, the while loop is executed n times, so that n comparisons are made, and therefore

$$W(n) = \max\{t(I) \mid I \in D_n\} = n$$

for $n \leq 1$.

Average Case Complexity:

Our random experiment consists of tossing a fair coin n times and observing the sequence of H's and T's that occur, the sample space for this random experiment is

$$S = \{(x_1, x_2, \ldots, x_n) \mid x_i = H \text{ or } T\}$$

and the events of interest are

$$E_i = \{(x_1, x_2, \ldots, x_n) \mid \text{first } H \text{ occurs at } x_i\}$$

for $1 \leq i \leq n$.

Now $s = (x_1, x_2, \ldots . x_n) \in E_i$ if and only if

$$s = (\underbrace{T, T, \ldots, T}_{i-1}, H, \underbrace{x_{i+1}, \ldots, x_n}_{n-i})$$

where $x_k \in \{H, T\}$ for $i + 1 \leq k \leq n$.

Since the tosses are independent and H and T are equally likely, then each of the outcomes in S is equally likely to occur, and therefore

$$p(E_i) = \text{prob(first } H \text{ is in position } i) = \frac{|E|}{|S|},$$

where $|S| = 2^n$ and $|E| = 2^{n-i}$, thus, $p(E_i) = \dfrac{2^{n-i}}{2^n} = \dfrac{1}{2^i}$.

If I_i represents the input that the first H occurs at the i^{th} position for $1 \leq i \leq n$, and I_{n+1} represents the input for which no H occurs, then the number of comparisons needed for the input I_i is $t(I_i) = i$, for $1 \leq i \leq n$, while $t(I_{n+1}) = n$, the average case complexity is therefore

$$A(n) = \sum_{i=1}^{n+1} p(I_i) \cdot t(I_i) = \sum_{i=1}^{n} \frac{i}{2^i} + \frac{n}{2^n}$$

Now, if $m(x) = x + 2x^2 + 3x^3 + \cdots + nx^n$, then we have

$$m(x) = \frac{x + nx^{n+2} - (n+1)x^{n+1}}{(1-x)^2}$$

for $x \neq 1$. Thus, if $x = \dfrac{1}{2}$, then

$$A(n) = \sum_{i=1}^{n} \frac{i}{2^i} + \frac{n}{2^n} = 2 - \frac{n+1}{2^n} < 2,$$

for $n \geq 1$. Therefore, on the average, less than 2 comparisons are made in order to find the location where the first H occurs.