Lecture - 19

Thursday, 8 September 2016 (15:20- 16:15)

Quick Sort Analysis

1 An Analogous Interesting Scenario

Imagine there are n shells as shown below. There are n shells kept between 2 poles as shown in the Figure 1

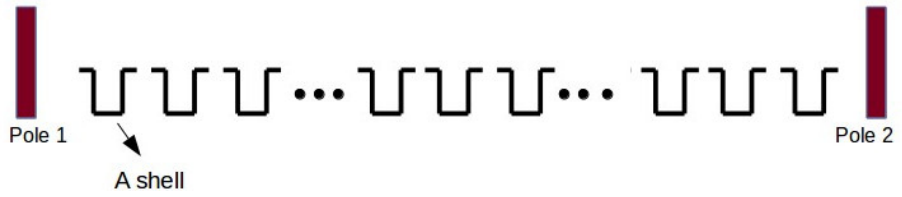

Fig. 1: The environment for the birds

A bird comes flying and lays egg in any of these shells as shown in Figure 2

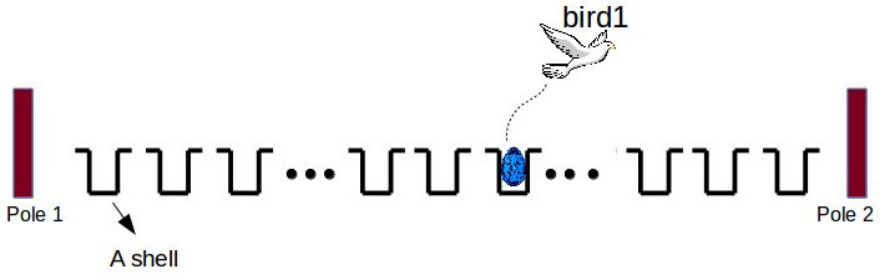

Fig. 2: Bird 1 hatching egg

Now, our entire region is divided into two sub-regions- one between the pole 1 and egg and another between the egg and the pole 2.

The next bird now comes and hatches eggs- **per sub-region one egg** as shown in Figure 3. Please note that the shells in which bird i has laid the eggs, are now on labelled by $B(i)$. Also, there can be only one egg per shell. So, a bird can not lay an egg in a shell which already has an egg.

Now, there are 4 sub-regions, where bird 3 comes and hatches eggs, one egg each sub-region as shown in Figure 4.

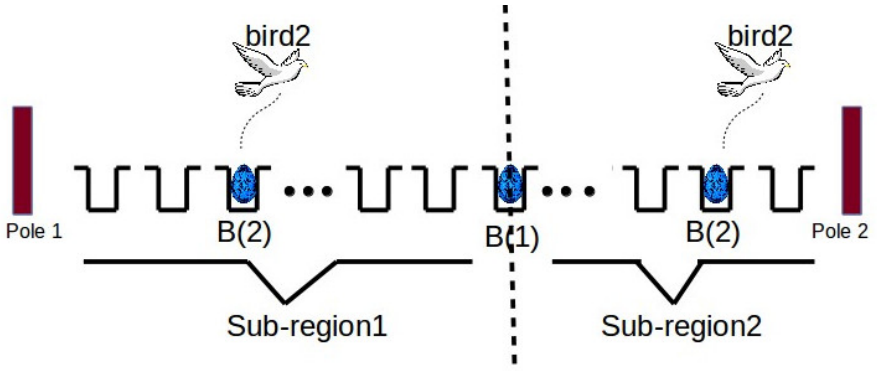

Fig. 3: Bird 2 hatching egg

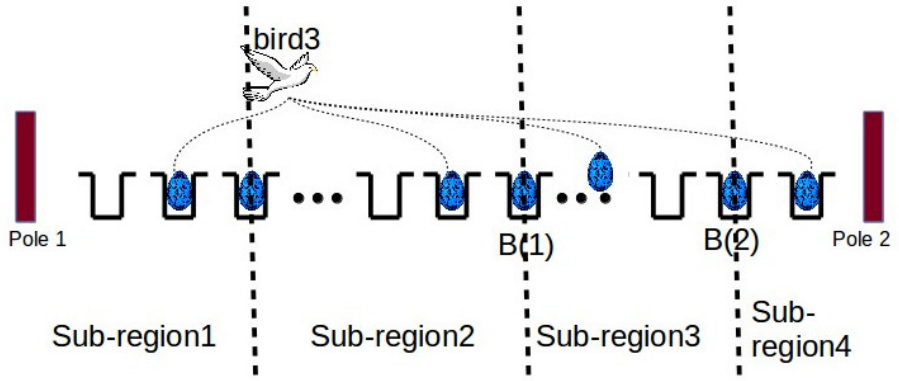

Fig. 4: Bird 3 hatching egg

Note: The wise reader can observe that the number of sub-regions do not double everytime. It depends on where the previous bird has layed its eggs.

We are interested in knowing how many birds should come and lay eggs such that all the shells are filled.

1.1 Worst Possible Case

Assume that each bird comes and hatches an egg in the first shell which is unfilled. Figure 5 shows what happens in such a scenario.

In this case, we need n birds to fill n shells.

Now, let us assume that our birds have grown smart and see what happens.

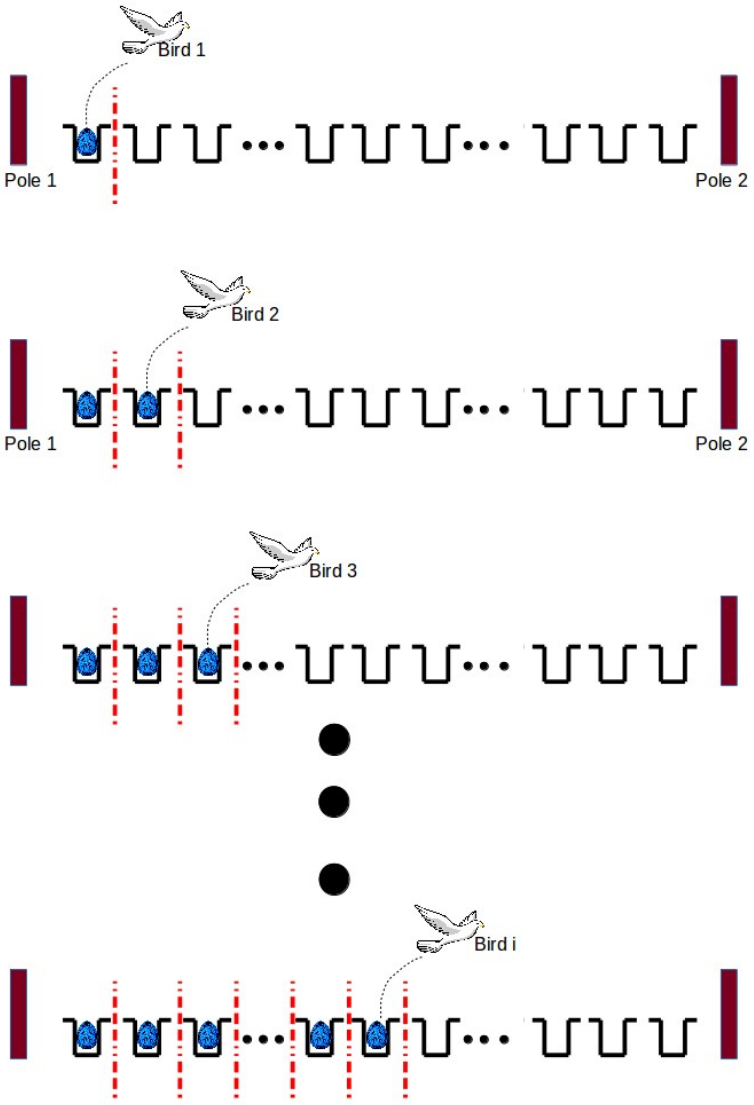

Fig. 5: Worst case scenario

1.2 Algorithm When the Birds are Smart

Assume that the birds are smart and they are prone to hatching when they are towards the center of a sub-region. So, for every sub-region, the bird hatches when it is towards the center of the sub-region. Assume that a sub-region runs from shell i to $i + k$, then the bird is likely to hatch in

one of the positions from $i + \frac{1}{4}k$ to $i + \frac{3}{4}k$. This is shown in Figure 6

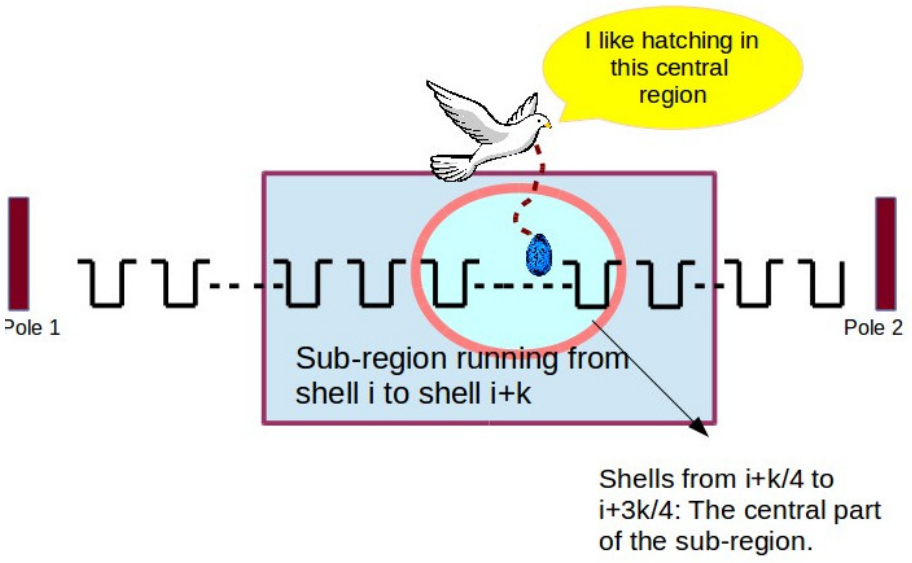

Fig. 6: The Smart Birds

How many birds do you think we need now?

Observe that when a bird divides a sub-region of size k, the size of any of the 2 sub-regions now created can not exceed $\frac{3}{4}k$. This is shown in Figure 7. We see from the Figure that, even in the worst case, if the bird chooses the boundary to hatch the egg, even then the maximum partition it can originate is $\frac{3}{4}$ of the original.

Hence, if there are n shells, the first bird can make a partition of maximum size $\frac{3}{4}n$.

Bird 2 can now create a maximum partition of $\frac{3}{4} \times \frac{3}{4} \times n$

.
.
.

Similarly, bird i can create a maximum partition of size $(\frac{3}{4})^i \times n$.

The process ends when a bird creates a maximum partition of size 1. Hence, putting

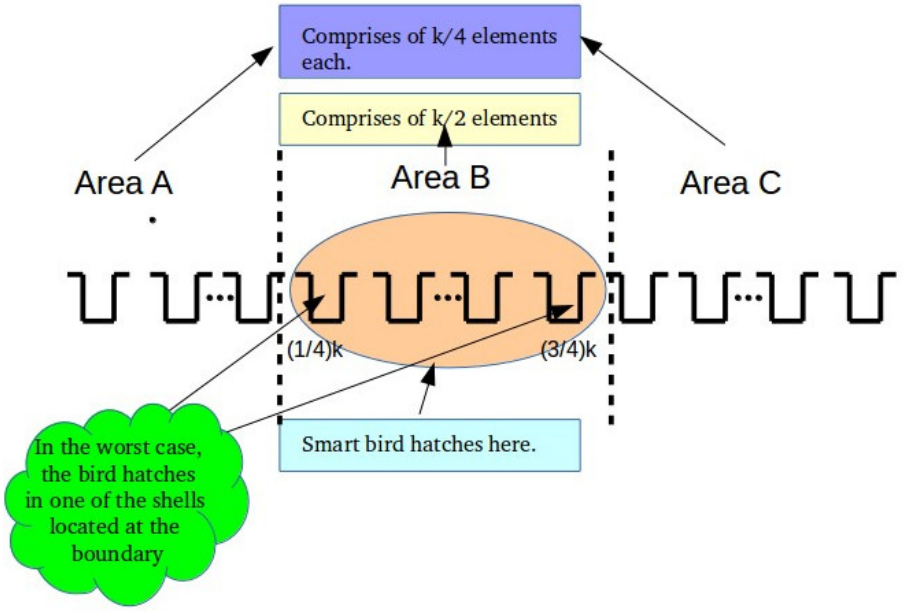

Fig. 7: The bigger division size can not exceed 3/4 of the original

$(\frac{3}{4})^i \times n \ = \ 1$

Take log both sides

$i \log \frac{3}{4} + \log n \ = \ \log 1$

$i \log \frac{3}{4} + \log n \ = \ 0$

$i \log \frac{3}{4} + = \ \log \frac{1}{n}$

$i \ = \ \frac{\log \frac{1}{n}}{\log \frac{3}{4}}$

Using, $\log_b a = \frac{\log_c a}{\log_c b}$

$i = \log_{\frac{4}{3}} n$

Hence, we need $\log_{\frac{4}{3}} n$ smart birds to fill all the n shells.

1.3 Analogy to the Quick Sort

If we are given a permutation of n numbers, if we randomly pick one of the numbers, the probability that it lies in the central part of the permutation is $\frac{1}{2}$. We have seen this in the previous lecture.

What we do is, pick a number r uniformly at random and check whether it lies in the central part of the sorted array. If yes, pick it, else repeat the experiment and pick another random number. On an average, one out of two attempts, you will get a number which lie in the central part of the sorted array.

Put the elements less than r towards its left and those greater than r towards its right. In doing so, r comes at the correct position. Look at Figure 8

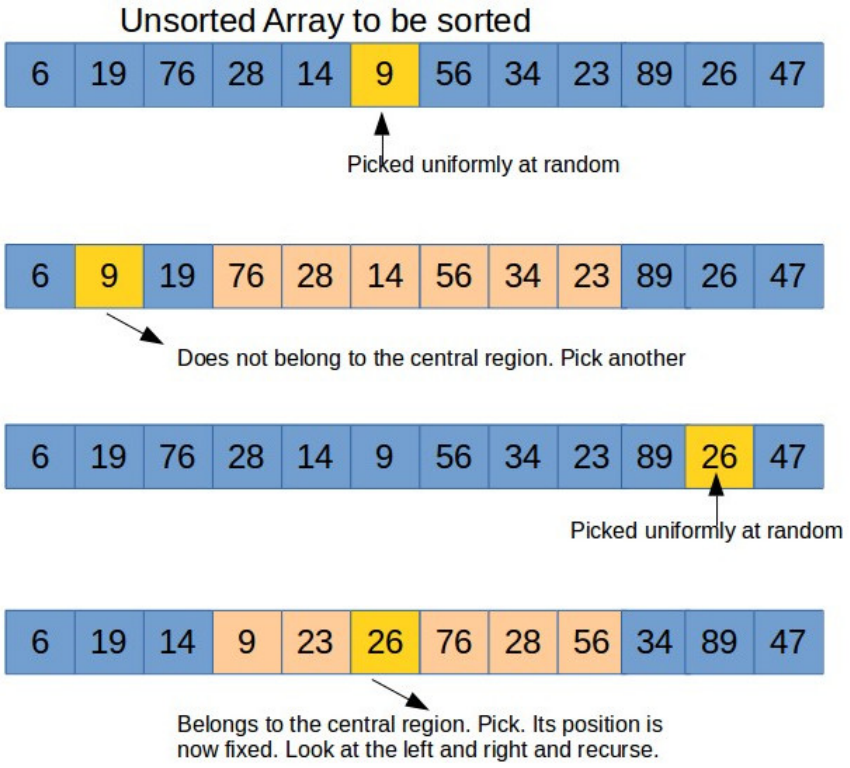

Fig. 8: Quick Sort

Can you now look at the analogy between the Quick Sort and the smart birds' game. The array has been now subdivided into 2 parts and since the picked part is in the central region, the maximum size of the sub-regions produced can be $\frac{3}{4}$ of the original.

See carefully what are we doing.

- We can pick an element which belongs to the central part of the array in a constant time. This is like the smart bird hatching an egg in the central part between the two poles. Just we take $O(n)$ extra steps to bring smaller elements to the left and larger to the right.
- We recurse the process on the left and right. Similarly, in the birds problem, the new bird comes and put an egg both on the left and right sides. Again, we take $O(n)$ extra steps to bring smaller elements to the left and larger to the right.

This has been illustrated in Figure 9.

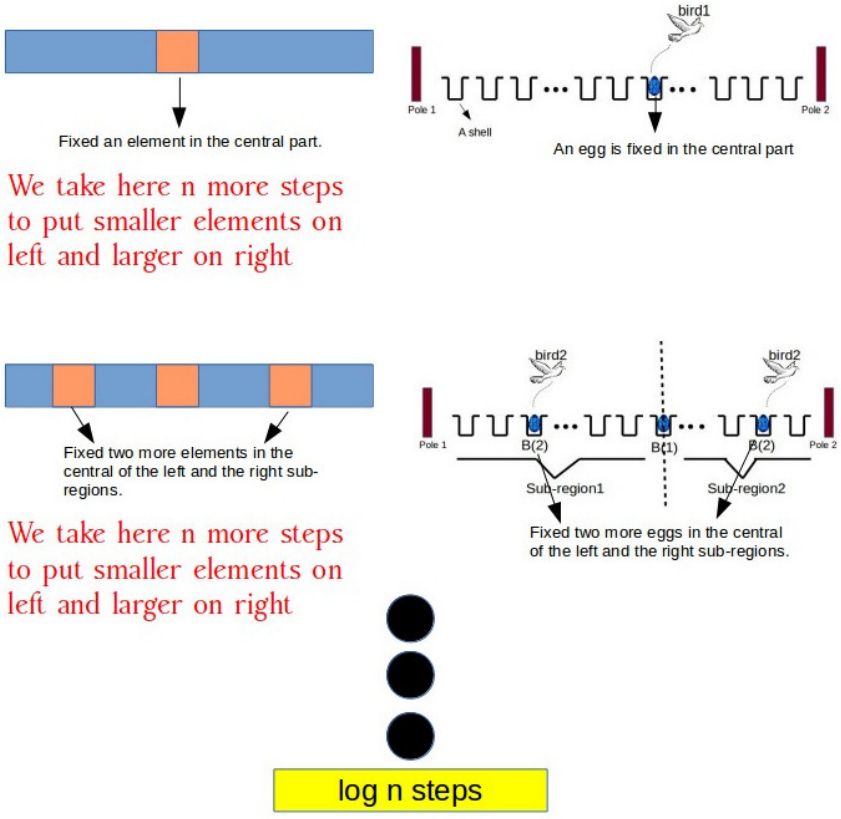

Fig. 9: Analogy between Quick Sort and Hatching Birds

Since, the quick sort algorithm is same as the hatching birds problem, except for the fact that, at every iteration, n more steps are taken in comparing the elements and putting them towards the left and right of the fixed element, the number of steps taken=

$$\log_{\frac{4}{3}} n + n \times \log_{\frac{4}{3}} n$$

$$=O(n \log_{\frac{4}{3}} n)$$