

### Problem 1:

Part a: Argue that the tail-recursive version of quicksort is correct. Lets do a proof by induction. It is clear that as a base case this algorithm sorts an array of size three correctly. Assume inductively that it sorts arrays of size  $3 \dots n - 1$  correctly. So after line 4, we have the invariant that the subarray  $A[0], \dots, A[q - 1]$  is sorted correctly, and followed correctly by  $A[q]$  (thanks to PARTITION - see page 146). Then the next time through the “while” loop, it runs on the array  $A[q + 1], \dots, A[r]$ , which again by induction is sorted correctly.

Part b: If we are unlucky in our choice of pivots, the size of the left subarray will be  $n - 1$  the first time, then  $n - 2$  the second time, and so on. The stack depth will be  $n - 1$ .

Part c: Instead of recurring on the left subarray every time, recur on the smaller of the two subarrays. In this way, if the size of the current subproblem is  $k$ , the size of the recursive subproblem is always  $\leq k/2$ , and after  $\lg n$  recursive calls we have a constant-size subproblem. Here is the new version of the code:

```
QUICKSORT( $A, p, r$ )
while ( $p < r$ ) do
     $q \leftarrow$  PARTITION( $A, p, r$ )
    if  $q - p < r - q$ 
        QUICKSORT( $A, p, q - 1$ )
         $p \leftarrow q + 1$ 
    else
        QUICKSORT( $A, q + 1, r$ )
         $r \leftarrow q - 1$ 
```

This algorithm continues to be correct, since both the left and the right subproblem get sorted, one by a recursive call and the other by the following iteration through the “while” loop.

### Problem 2:

PERMUTE-BY-CYCLIC takes element  $A[i]$  to element  $B[j]$  if  $j = (i + \text{offset}) \bmod n$ , which is true if  $\text{offset} \bmod n = (j - i) \bmod n$ . Since  $\text{offset}$  is chosen from  $\{0, \dots, n - 1\}$ , there is only one value for  $\text{offset}$  which satisfies this equation, and that one value is chosen with probability  $1/n$ . But the resulting permutation is not uniformly random. For instance, PERMUTE-BY-CYCLIC can never produce the permutation in which  $B[0] = A[1], B[1] = A[0]$ , and for all other  $i \in \{2, \dots, n - 1\}, A[i] = B[i]$ .

### Problem 3:

The sample space in this problem is the set of possible permutations of hats. The relevant events are the events that customer  $i$  gets his own hat. Call this  $e_i$ . Since customer  $i$  gets a random hat,  $\Pr[e_i] = 1/n$ . The random variable we are interested in is the number of customers who get their own hats back. Call this  $V$ . We have

$$V = \sum_{i=1}^n V_i$$

where  $V_i$  is an indicator variable for  $e_i$ , that is,  $V_i = 1$  when customer  $i$  gets his hat back. The problem asks for the expectation of  $V$ :

$$E[V] = \sum_{i=1}^n E[V_i] = \sum_{i=1}^n \Pr[e_i] \sum_{i=1}^n 1/n = 1$$

### Problem 4:

First of all, the problem is hard to decipher. So let's review what it is asking for. We have a "probabilistic counter", controlled by an increasing series of integers  $n_0, n_1, n_2, \dots$ . The counter begins by containing  $n_0$ , and always contains one of the  $n_i$ . When INCREMENT is called, if the counter contains  $n_i$ , it changes from  $n_i$  to  $n_{i+1}$  with probability  $1/(n_{i+1} - n_i)$ , and otherwise it stays the same.

So how do we expect this little machine to behave? Calling INCREMENT is like rolling a die with probability  $1/(n_{i+1} - n_i)$  of coming up one; we know that the expected number of rolls required to get a one is  $n_{i+1} - n_i$ . So for instance if the  $n_i$  were 0, 3, 22, 25, 100, we expect to have to call INCREMENT three times before the counter switches from zero to three, 19 times before the counter switches from 3 to 22, three times before the counter switches from 22 to 25, and so on. So it seems like when the number of INCREMENT calls is exactly one of the  $n_i$ , that is about when the counter should be switching to that  $n_i$ .

Now we are asked to prove that after calling INCREMENT  $n$  times, the expected value of the counter is  $n$ . Notice that when  $n$  is not one of the  $n_i$ , the probability that the counter actually contains  $n$  is zero; the counter only has non-zero probabilities of containing the  $n_i$ , and those probabilities change every time INCREMENT is called. If we knew those probabilities we could compute the expectation of the value in the counter.

Call the random variable for the value of the counter after  $n$  INCREMENT operations  $V_n$ . And let  $d_k = n_{k+1} - n_k$ , so that when the counter contains  $n_k$  the probability that it changes when an INCREMENT operation occurs is  $1/d_k$ .

Let  $e_{n,k}$  be the event that after  $n$  INCREMENT operations the counter contains value  $n_k$ . This is a useful event because

$$E[V_n] = \sum_{k=1}^n \Pr[e_{n,k}] n_k$$

Clearly when  $n = 0$ ,  $V_n = 0$  so  $E[V_n] = 0$ . So we have a base case, and if we can figure out how  $E[V_n]$  changes as  $n$  increases, we can use induction to get a formula for any  $E[V_n]$ .

Some of the probabilities  $Pr[e_{n,k}]$  increase and some decrease as  $n$  goes to  $n + 1$ . Let's try and get a formula for how they change. Observe that  $e_{n+1,k}$  can occur in only two ways. The first way is that the counter already contained value  $n_k$  after  $n$  INCREMENTS, and the last INCREMENT failed to change the counter value. The probability of this is  $Pr[e_{n,k}](1 - 1/\delta_k)$ . The other possibility is that the counter contained value  $n_{k-1}$  after  $n$  INCREMENTS, and the last INCREMENT changed the counter value from  $n_{k-1}$  to  $n_k$ . The probability of this is  $Pr[e_{n,k}](1/\delta_{k-1})$ . So

$$\begin{aligned} \Pr[e_{n+1,k}] &= \Pr[e_{n,k}](1 - 1/\delta_k) + \Pr[e_{n,k-1}](1/\delta_{k-1}) \\ &= \Pr[e_{n,k}] - \Pr[e_{n,k}](1/\delta_k) + \Pr[e_{n,k-1}](1/\delta_{k-1}) \end{aligned}$$

And we can write

$$\begin{aligned} E[V_{n+1}] &= \sum_{k=1}^n n_k (\Pr[e_{n,k}]n_k - \Pr[e_{n,k}](1/\delta_k) + \Pr[e_{n,k-1}](1/\delta_{k-1})) \\ &= \sum_{k=1}^n \Pr[e_{n,k}]n_k + \sum_{k=1}^n \Pr[e_{n,k-1}](1/\delta_{k-1})n_k - \Pr[e_{n,k}](1/\delta_k)n_k \end{aligned}$$

By the inductive assumption, the first sum in the expression above is equal to  $n$ . If we imagine writing out the second sum, we'd get something like this:

$$\begin{aligned} \Pr[e_{n,0}](1/\delta_0)n_1 - \Pr[e_{n,1}](1/\delta_1)n_1 + \Pr[e_{n,1}](1/\delta_1)n_2 - \Pr[e_{n,2}](1/\delta_2)n_2 + \Pr[e_{n,2}](1/\delta_2)n_3 - \Pr[e_{n,3}](1/\delta_3)n_3 + \\ \dots - \Pr[e_{n,n}](1/\delta_n)n_n + \Pr[e_{n,n}](1/\delta_n)n_{n+1} - \Pr[e_{n,n+1}](1/\delta_{n+1})n_{n+1} \end{aligned}$$

Since  $\delta_0 = n_1$ , the first term in this sum is:

$$\Pr[e_{n,0}](1/\delta_0)n_1 = \Pr[e_{n,0}](1/n_1)n_1 = \Pr[e_{n,0}]$$

The last term in the big sum is zero, since  $e_{n,n+1}$ , the event that the counter value is  $n_{n+1}$  after only  $n$  INCREMENTS, which is impossible. The other terms we can pair up:

$$\begin{aligned} -\Pr[e_{n,1}](1/\delta_1)n_1 + \Pr[e_{n,1}](1/\delta_1)n_2 - \dots - \Pr[e_{n,n}](1/\delta_n)n_{n-1} + \Pr[e_{n,n}](1/\delta_n)n_{n-1} \\ = \Pr[e_{n,1}](1/\delta_1)(n_2 - n_1) + \Pr[e_{n,2}](1/\delta_2)(n_3 - n_2) + \dots + \Pr[e_{n,n}](1/\delta_n)(n_n - n_{n-1}) \\ = \Pr[e_{n,1}] + \Pr[e_{n,2}] + \dots + \Pr[e_{n,n}] \end{aligned}$$

Since  $n_{k+1} - n_k = \delta_k$ , we have

$$E[V_{n+1}] = E[V_n] + \sum_{k=0}^n \Pr[e_{n,k}] = E[V_n] + 1$$

The last step follows since the counter has to contain one of the values  $n_0, \dots, n_n$  after  $n$  INCREMENTS. Since  $E[V_0] = 0$ , we see by induction that  $E[V_n] = n$ .