Zonal Informatics Olympiad, 2002, Part A

Model Solutions

1. (a) Find the largest disk and flip the top of the stack upto and including this disk. This will take the largest disk to the top of the pile. Then flip the entire stack to get the largest disk to the bottom.

At step k+1, inductively the k largest disks are arranged in order at the bottom of the stack. Pick the largest disk above these k disks, flip the stack above this disk to take it to the top and then flip the top n-k disks to bring it to position k+1 from the bottom.

(b) Let the disks be numbered 1,2,...,n, where disk 1 is smaller than disk 2 is smaller than ... disk n. In the worst case it takes two flips for each of the disks n, n-1, n-2, ...,3. After disks 3 to n are in the correct position, in the worst case 1 and 2 are inverted so we need one more flip. So, in the worst case, we need 2(n-2)+1 = 2n-3 flips.

```
For 2 disks, the worst case is   2
                                 1
Call this arrangement s2
```

For 3 disks, the worst case is to get to s2 after 3 comes to the bottom. We would like to take 2 flips to get 3 to the bottom, so, working backwards,

```
2         3         1
1   <-    1    <-   3  = s3
3         2         2
```

For 4 disks, we want s1 above 4 after flipping twice to get 4 to the bottom, so s4 is the last sequence of the following

```
1         4         3
3   <-    2    <-   2  = s4
2         3         4
4         1         1
```

For k+1 disks, we want sk above disk k+1 after flipping twice to get k to the bottom, so s(k+1) is obtained as follows: (Notation: reading a sequence from top to bottom, head is the top element, tail is the rest, last is the bottom element, init is all but the bottom, flip is a function that inverts a list):

```
sk           (k+1)       flip (init(flip sk)) = tail sk
:      <-      :      <-    (k+1)                (k+1)
(k+1)      flip sk       last (flip sk)        head sk
```

2. Formally:

Conditional probability (Bayes):

Pr(A given B) = Pr(A and B) / Pr (B)

(a) B is "test is positive", A is "has TB"

```
Pr (A and B) is (80/100)*(10/100) = 8/100
Pr (B) is Pr(yes TB, test positive) + Pr(no TB, test positive)
         = (10/100)*(80/100)        + (90/100)*(20/100)
         = 26/100
```

So Pr(A given B) is 8/26 = 4/13

(a) B is "test is negative", A is "has TB"

Using similar computation as in (a)

Pr (A and B) is 2/100
Pr (B) is 74/100
Pr(A given B) 2/74 = 1/37

Informally:

In 100 people, 10 have TB, 90 do not.
For the 10 who do, 8 test positive, 2 do not.
For the 90 who don't, 18 test positive, 72 do not.

(a) 8 of 26 people who test positive have TB
(b) 2 of 74 people who test negative have TB

3. Think of one pan as the 'positive' pan and the other as the 'negative' pan. If we put w1 in the positive pan and w2 in the negative pan, the net weight is w1 - w2. Thus, if we use weights w1, w2, ..., wn, we can write out the net weight as a "ternary" n-digit number using digits +1/0/-1.

From this, it follows that the optimal set of weights corresponds to powers of 3.

(a) 1, 3, 9, 27, 81, 243, 729, 2187, 720 (the last weight is 4000-3280, where 3280 is the sum of the previous 8 weights).

(b) In general, we would use 1, 3, 9, ..., log_3(n), n-(1+3+..+log_3(n))

4. (a) Look at A after rearranging the columns.

Consider any pair of adjacent columns k-1 and k. We show that for each row j, A(j,k-1) is smaller than A(j,k).

The first entry A(1,k) is bigger than A(1,k-1). This is because A(1,k) dominates at least one entry in column k-1 (the one that was originally to the left of it) and so dominates A(1,k-1), the minimum element in column k-1.

Now consider A(2,k). We can argue that A(2,k) dominates at least two entries in column k-1 and hence dominates A(2,k-1), the second smallest entry in column k-1.

Suppose A(2,k) dominates only one entry in column k-1 (it dominates at least one, the one originally to its left). Then, the element originally to the left of A(2,k) is the element currently at A(1,k-1). This means that A(1,k) originally had another element A(j,k-1) to its left, not A(1,k-1). But then, A(2,k) > A(1,k) > A(j,k-1) > A(1,k-1), which contradicts the assumption that A(2,k) dominates only one entry in column k-1.

In a similar manner, we can show that for each j, A(j,k) dominates at least j entries in column k-1 and so dominates A(j,k-1).

Since for each row j and each column k, A(j,k) is bigger than A(j,k-1), it follows that all the rows are still in ascending order.

(b)(i) Check k against the top right corner element A(1,n)

Case 1: If k is smaller than A(1,n) then k cannot appear in the last column (since A(1,n) is the smallest entry in that column) so we have to search for A in rows 1 to n, columns 1 to n-1.

Case 2: If k is bigger than A(1,n) then k cannot appear in the first row (since A(1,n) is the largest entry in that row) so we have to search for A in rows 2 to n, columns 1 to n.

In the next step, we check k against the top right corner of the remaining matrix of interest (the top right corner is A(1,n-1) if Case 1 holds and A(2,n) if Case 2 holds) and repeat the earlier analysis.

With each step we reduce the matrix of interest by pruning either one row or one column. Since we started with n rows and n columns, within 2n steps (actually 2(n-1) steps) we would have pruned the matrix down to a single row and column, i.e. a single element.

(ii) Since the search proceeds from the top right to the bottom left, the search goes on longest if the element is at the bottom left corner, A(n,1).

5. (a) There are two natural methods:

The first is to pick up a pair of balls, weigh them, retain the heavier. Then, pick up each of the other balls in turn, weigh them against the current "champion" and retain the heavier one as the new "champion". The ball that remains at the end is the heaviest.

The second method is to organize a knockout tournament. Pair up all the balls and retain the heavier one from each pair. Then, pair up the winners of the first round and do the same. Repeat until a single winner is found.

Both methods require n-1 weighings.

(b) The tournament solution to (a) helps us find the second heaviest ball more efficiently than by naively repeating the procedure for (a).

In the overall tournament, the second heaviest ball would have beaten every ball other than the heaviest one. So, we only have to find the heaviest among the log n balls that "lost" to the heaviest ball to find the second heaviest.

6. In each round, the number of black balls can either decrease by 1 (if we pick up black-black or black-white) or increase by 1 (if we pick up white-white) so not much information can be gleaned from analysing the black ball count.

However, notice that white balls are eliminated in pairs. Thus, the parity (even/odd) of the white balls in the bag is an invariant. Eventually, we have a single ball (odd). Thus, the last ball is white if only if the bag originally had an odd number of white balls.

==