Final Exam Solutions for CS 172, Spring '99

Professor Manuel Blum

Problem 1

Q: What base?
A: 2

Q1.1:
n = 1 + [$\log_2$ N]

Q1.2:
a. (n + 1)/2
b. O(2^n * n) or Omega(2^n * n)

Q1.3:
(a) is poly(n) and (b) is exp(n)

Problem 2

Q2.1:
Move disks 1...n from A via B to C:
If n=1, move disk 1 from A to C; else

1. move disks 1...n-1 from A via C to B;
2. move disk n from A to C
3. move disks 1...n-1 from B via A to C.

Q2.2:
f(n) = f(n-1) + 1 + f(N-1) = 2f(N-1) = 1 = 2^n - 1

Q2.3:
No. To move disk n to C, it is necessary that disks 1...n-1 be on B (>=f(n-1) steps). After disk n is moved to C (>=1 step), it is necessary to move all disks 1..n-1 from B to C (>=f(n-1) steps). Thus the algorithm takes >=f(n) = 2^n-1 steps.

Problem 3

Q3.1:
Depends: Yes if all pieces of furniture are in correct order. No otherwise.

Q3.2:
Yes. Suppose wlg that the final order is [5 6 _ _ (new line) 1 2 3 4]. First rotate clockwise 1 into correct position (this needs just 1 empty slot). This takes O(n^2) steps. Then rotate 2 into position diagonal to 1: [_ 2

... (new line) 1 ...] by rotating clockwise. Then put 2 in its correct position. This reduces the 2 x n problem to a 2 x (n-1) problem with 2 empty slots. Proceed inductively.

Q3.3:
Yes. Suppose wlg that the final order is [9 10 _ _ (new line) 5 6 7 8 (new line) 1 2 3 4]. This is the case m = 3, n = 4.

1. Use the two empty slots to move 1 to a position against the wall. Then rotate 1 into position (just 1 empty slot against the wall suffices to do this.
2. Put an empty slot adjacent to 1 in the position slated for 2, and rotate 2 into a position next to that slot.
3. Continue recursively to get [(junk)...bottom row: 1 2 3 4].
4. Now use method of Q3.2 to finish the job on the 2 remaining rows.

Q3.4:
Yes. Use the same method as Q3.3 to get the bottom first row correct and then recursively get the remaining rows correct. Notice that this takes O(n^3) steps for each of m recursions, or O(mn^3) total.

Problem 4

Q4:
Because there are only a finite number of possible configurations, one can conduct (in principle) a graph whose nodes are configurations, edges join 2 configurations if one can go from one configuration to the other in just one step. Consequently, this is decidable. I don't know the answer to any of the other questions. The special case of problem 3 is in P.