CS 577: Introduction to Algorithms 9/4/2012

Homework 0

Instructors: Shuchi Chawla and Dieter van Melkebeek

This is a self-calibration assignment. You should solve the problems to refresh some of the pre-requisites for the course, and to get an idea of the level of difficulty. This assignment will not be graded but model solutions will be handed out in class on 9/6.

1. Review Chapter 2 of the course text. After doing so, you should know things like how to implement a priority queue using a heap and how to construct this representation for a given sequence of n elements using $O(n)$ elementary operations.

2. Consider the following algorithm.

```
int square (int x) {
  return x*x;
}

int fastExp (int a, int b) {

  // input: b >= 0
  // output: a raised to the power b

  if (b == 0) {
    return 1;
  }
  else if (b % 2 == 0) {
    return square(fastExp(a,b/2));
  }
  else return a*fastExp(a,b-1);
}
```

   (a) Give a correctness proof for `fastExp`, i.e., prove that for any integers a and b with $b \geq 0$, `fastExp(`a`,`b`)` returns a raised to the power b. Hint: Use induction. Clearly specify the induction hypothesis.

   (b) How many recursive calls will be made in case b is of the form $b = 2^k$ for integer k? How often will two elements be multiplied?

   (c) How do the answers to the previous question change when you replace the line

   ```
   return square(fastExp(a,b/2));
   ```

   with

   ```
   return fastExp(a,b/2)*fastExp(a,b/2);
   ```

3. A *tournament* is a directed graph which contains exactly one of the two possible edges between each pair of vertices. Think of the vertices as players; every vertex plays every other vertex once. An edge (u, v) indicates that u beat v in their game; there are no ties.

A *dominating set* in a directed graph is a set of vertices D such that every vertex v outside D is beaten by at least one vertex in D.

Show that every tournament with n vertices has a dominating set D of size $O(\log n)$. Give a polynomial-time algorithm to find such a set.

4. Consider the traditional marriage algorithm we discussed in class. Construct an example in which the following scenario happens: A girl prefers b_1 over b_2, and b_2 over b_3, and gets married to b_2; when she switches b_2 and b_3 in her ranking, she gets married to b_1.

   This example shows that a girl may get married to a someone she likes better by lying about her preferences. Argue that such a strategy does not work for a boy, i.e., by lying about his preference between two girls g_2 and g_3, he cannot hope to get married to a girl g_1 whom he (truly) ranks higher than both g_2 and g_3.

5. Consider the following puzzle. You are given a sequence of m black discs and n white discs on an oval-shaped track with a turnstile capable of flipping (i.e., reversing) three consecutive discs. In Figure 1, there are 8 black discs and 10 white discs on the track. You may spin the turnstile to flip the three discs in it or shift one position clockwise for each of the discs on the track (see Figure 1). The goal of the puzzle is to gather the discs of the same color in adjacent positions using flips and shifts (see Figure 2).

   Determine as a function of m and n the sequences for which the puzzle is solvable.

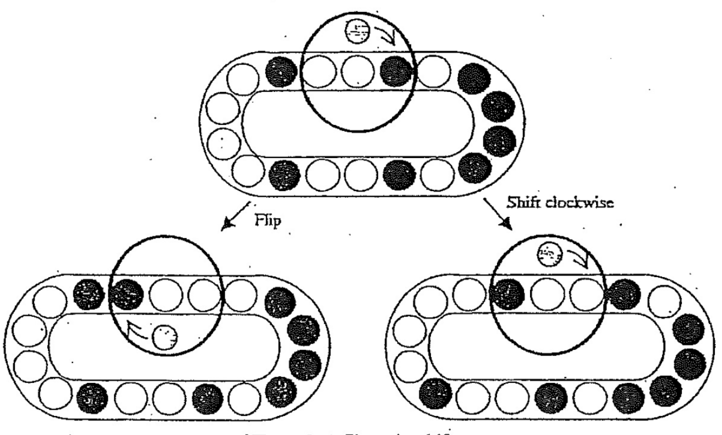

Figure 1. A flip and a shift

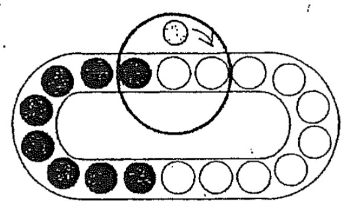
Figure 2. A goal sequence