15-750 Algorithms (Spring 2002) **Prof. Manuel Blum**

Homework #3 **due: Feb. 8 (in class)**

(problems 1 & 2 are due to Danny Sleator)

Problem 1: Path reversal in Union-Find (20 points)

Recall from the lecture on Union-Find the process of *path compression*: after finding the root v of the tree containing u in a `find(u)`, we traverse the path from u to v one more time and change the parent pointers of all vertices along the path to point directly to v (Figure 1). This technique (together with making the root of the smaller tree a child of the

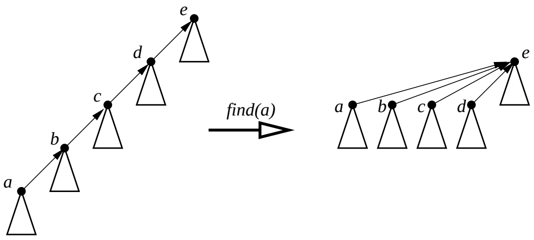

Figure 1: Path compression.

root of the larger in `union` operation) allowed a sequence of m `union` and `find` operations starting with n singleton sets to be performed in $\mathcal{O}((m+n)\alpha(n))$ steps, where $\alpha(n)$ is the inverse of Ackermann's function.

One drawback of such path compression is the fact, that the path from u to v is traversed twice. Consider the following variation of the algorithm, called *path reversal*: traverse the path only once, and make every node on the path point to the *first* node on the path (Figure 2). Note that this changes the root of the tree. Is the method of path reversal

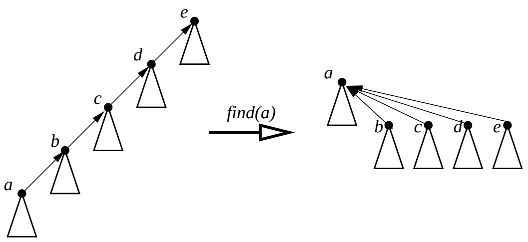

Figure 2: Path reversal.

a worthwhile modification to Union-Find, i.e., can we still guarantee the $\mathcal{O}((m+n)\alpha(n))$ bound? Justify your answer.

Hint: Think binomial tree.

Problem 2: Quads and Blobs (30 points)

Consider a very reactive elementary particle called a quad. Quads react with one another to form blobs of various sizes. A blob results when two quads collide, or when a quad and a blob collide. (When two blobs collide they bounce apart unaffected.) The size of a blob is the number of quads it has absorbed. (The smallest blob has size two.) A blob of size 5 is unstable and instantaneously explodes into two blobs of size 2, a gamma ray, and two new quads. The following reactions summarize the situation: (Q is a quad, and B_i is a blob of size i.)

$$Q + Q \to B_2$$

$$Q + B_2 \to B_3$$

$$Q + B_3 \to B_4$$

$$Q + B_4 \to 2B_2 + \gamma + 2Q$$

A quiescent configuration is one containing no quads. Suppose that into a quiescent configuration a new quad is introduced. This causes a chain reaction that produces a number of gamma rays and new quads. Eventually a new quiescent state is reached.

1. Choose a constant c and a potential function (which depends on the number of various particles present). Prove the following three statements about these choices:
   (a) The amortized number of gamma rays produced when a new quad is introduced is at most c.
   (b) Starting from an empty configuration, the introduction of k quads causes the emission of at most ck gamma rays.
   (c) Show that your value of c is the smallest one satisfying parts (a) and (b).
2. *(Extra credit)* Suppose you were given a set of n different types of particles (one of which is the quad), and m reactions involving these particles. The left hand side of each reaction is a combination of particles required for it to occur, and the right hand side is the outcome of the reaction. Each reaction also produces zero or more gamma rays. How would you approach the problem of determining if an infinite chain reaction is possible? If such a reaction is impossible, how would you determine the minimum constant c satisfying parts 1a and 1b above.

Problem 3: Variations on Splay Trees (extra extra credit)

Recall the rules of `splay(`x`, `S`)`:

(i) if x has a parent but no grandparent, just `rotate(`x`)`

(ii) if x has a parent y and a grandparent, and if x and y are either both left children or both right children, first `rotate(`y`)`, then `rotate(`x`)`

(iii) if x has a parent y and a grandparent, and if one of x, y is a left child and the other is a right child, first `rotate(`x`)`, then `rotate(`x`)` again

A smart programmer wishes to simplify an implementation of splay trees, and proposes three new sets of the rules:

1. replace rules (ii) and (iii) by:
   if x has a parent y and a grandparent, first `rotate(`x`)`*, then* `rotate(`x`)` again
   (this case was shortly discussed in the lecture)

2. replace rules (ii) and (iii) by:
   if x has a parent y and a grandparent, first `rotate(`y`)`*, then* `rotate(`x`)`

3. replace rule (iii) by:
   if x has a parent y and a grandparent, and if one of x, y is a left child and the other is a right child, just `rotate(`x`)`

Discuss the consequences of each variation on the performance. Two performances are considered roughly "the same", if each is $\mathcal{O}(\cdot)$ of the other. Justify your claims with analytic arguments.