Episode 55: Exercises rev. 17/04/2008 08:36

Ingredients: Amortization and rehashing

Exercise Session

Consider a hash table with a certain size, N. In this exercise we will deduce a good strategy for "rehashing" in a dynamic hashing scheme. Let the term *fill ratio* denote the threshold where the hash table are expanded and let *scale factor* denote the factor that the size is scaled with upon expansion. If, for example, the fill ratio is 0.5 and the scale factor is 2, we will perform a rehashing in to a new hash table of size $2N$ as soon we reach $0.5N$ elements in current table. Assume that the cost of initializing a new memory cell is c_i and the cost of adding a new element to the hash table is c_a.

1. Calculate the cost of inserting 50 elements into an initially empty hash table with an initial size of 4. Perform the calculation where "fill ratio" is either 0.2 or 1.0 and "scale factor" is either 1.5 or 3 (notice: there are four calculations to perform).
2. What would you choose as "fill ratio" and "scale factor"? Why?
3. Most hash implementations use a "fill ratio" of 0.5 and a "scale factor" of 2. But apparently these values does induce the minimum cost of insertion. Why are they selected anyway?

A Problem For Self-Study

Would it be more economical to expand a hash table by tripling (rather than doubling) the table in size when the table is half full? Address this question by performing an asymptotic amortized analysis of both cases for insertion of N elements into an initially empty hash table. Assume constants c_a and c_r as above. You should obtain two formulae parameterized by c_a and c_r. Then discuss what kind of general conclusions can be made from these formulae.

Warning: This is a relatively difficult problem