CSI 3334, Data Structures

Lecture 7: Time Complexity of Skip List

Date: 2012-9-11
Author(s): Robert Bissell, Charles Stokes
Lecturer: Fredrik Niemelä

This lecture explains the run time of a skip list. Our analyzing the formula behind the skip list will help to determine if it is a more cost efficient data structure to use.

1 What is a Skip List?

The skiplist is a data structure that stores heiarchical lists that connect increasingly sparse lists of data. The sparseness of each "level" is indicated by a probability that makes each subsequent level less likely to be sparse.

1.1 Analysis

Define Variables:
n = number of nodes
p = probability that an element in height i is also in height i+1
k = current level
For analysis, assume that there is no maximum height. In practice, this must be done in implementation.

1.1.1 Determine the number of pointers on average per element

$$\frac{\sum np^{k-1}}{n} = 1 + p + p^2 + p^3 + ... = \frac{1}{1-p}$$

Element $\frac{1}{p}$is the last useful level (the next level has only one element). Solve to find L(n), the upper expected maximum height.

$$\frac{1}{p} = np^{k-1}$$

$$np^k = 1$$

$$n = (\frac{1}{p})^k$$

$$k = log_{\frac{1}{p}} n = L(n)$$

1.1.2 Analyze Search

Search "feels" recursive, ie the amount of work left to do per step after each step decreases.

Essentially, we are looking for the expected length of a search path.

define cost c

$c(k)$ =expected length of a search path

$$c(0) = 0$$

Assuming probalistically independent:

$$c(k) = p(1 + c(k-1) = (1-p)(1 + c(k))$$

$$c(k) = p + pc(k-1) + 1 - p + c(k) - pc(k)$$

$$c(k) = c(k-1) + \frac{1}{p}$$

$$= \frac{k}{p} + c(0) = \frac{k}{p}$$

Runtime:

$$\frac{L(n) - 1}{p} + \frac{1}{p} = \frac{L(n)}{p}$$