76], [Clapson, P., 77], [Reingold, E.M. *et al.*, 77], [Rosenberg, A.L. *et al.*, 77], [Gotlieb, C.C. *et al.*, 78], [Guibas, L.J., 78], [Halatsis, C. *et al.*, 78], [Kollias, J.G., 78], [Kronsjo, L., 79], [Mendelson, H. *et al.*, 79], [Pippenger, N., 79], [Romani, F. *et al.*, 79], [Scheurmann, P., 79], [Larson, P., 80], [Lipton, R.J. *et al.*, 80], [Standish, T.A., 80], [Tai, K.C. *et al.*, 80], [Bolour, A., 81], [Litwin, W., 81], [Tsi, K.T. *et al.*, 81], [Aho, A.V. *et al.*, 83], [Nishihara, S. *et al.*, 83], [Reingold, E.M. *et al.*, 83], [Larson, P., 84], [Mehlhorn, K., 84], [Torn, A.A., 84], [Devroye, L., 85], [Szymanski, T.G., 85], [Badley, J., 86], [Jacobs, M.C.T. *et al.*, 86], [van Wyk, C.J. *et al.*, 86], [Felician, L., 87], [Ramakrishna, M.V., 87], [Ramakrishna, M.V. *et al.*, 88], [Ramakrishna, M.V., 88], [Christodoulakis, S. *et al.*, 89], [Manber, U., 89], [Broder, A.Z. *et al.*, 90], [Cormen, T.H. *et al.*, 90], [Gil, J. *et al.*, 90].

3.3.1 Practical hashing functions

For all the hashing algorithms we assume that we have a hashing function which is 'good', in the sense that it distributes the values uniformly over the table size range m. In probabilistic terms for random keys k_1 and k_2 this is expressed as

$$Pr\{h(k_1) = h(k_2)\} \leq \frac{1}{m}$$

A **universal class of hashing functions** is a class with the property that given any input, the average performance of all the functions is good. The formal definition is equivalent to the above if we consider h as a function chosen at random from the class. For example, $h(k) = (ak + b) \bmod m$ with integers $a \neq 0$ and b is a universal class of hash functions.

Keys which are integers or can be represented as integers, are best hashed by computing their residue with respect to m. If this is done, m should be chosen to be a prime number.

Keys which are strings or sequences of words (including those which are of variable length) are best treated by considering them as a number base b. Let the string s be composed of k characters $s_1 s_2 \ldots s_k$. Then

$$h(s) = \left(\sum_{i=0}^{k-1} B^i s_{k-i} \right) \bmod m$$

To obtain a more efficient version of this function we can compute

$$h(s) = \left(\left(\sum_{i=0}^{k-1} B^i s_{k-i} \right) \bmod 2^w \right) \bmod m$$

where w is the number of bits in a computer word, and the $\bmod 2^w$ operation is done by the hardware. For this function the value $B = 131$ is recommended, as B^i has a maximum cycle $\bmod 2^k$ for $8 \leq k \leq 64$.

Hashing function for strings

```
int hashfunction(s)
char *s;

{ int i;
  for(i=0; *s; s++) i = 131*i + *s;
  return(i % m);
}
```

References:
[Maurer, W.D., 68], [Bjork, H., 71], [Lum, V.Y. *et al.*, 71], [Forbes, K., 72], [Lum, V.Y. *et al.*, 72], [Ullman, J.D., 72], [Gurski, A., 73], [Knuth, D.E., 73], [Lum, V.Y., 73], [Knott, G.D., 75], [Sorenson, P.G. *et al.*, 78], [Bolour, A., 79], [Carter, J.L. *et al.*, 79], [Devillers, R. *et al.*, 79], [Wegman, M.N. *et al.*, 79], [Papadimitriou, C.H. *et al.*, 80], [Sarwate, D.V., 80], [Mehlhorn, K., 82], [Ajtai, M. *et al.*, 84], [Wirth, N., 86], [Brassard, G. *et al.*, 88], [Fiat, A. *et al.*, 88], [Ramakrishna, M.V., 88], [Sedgewick, R., 88], [Fiat, A. *et al.*, 89], [Naor, M. *et al.*, 89], [Schmidt, J.P. *et al.*, 89], [Siegel, A., 89], [Mansour, Y. *et al.*, 90], [Pearson, P.K., 90], [Schmidt, J.P. *et al.*, 90].

3.3.2 Uniform probing hashing

Uniform probing hashing is an open-addressing scheme which resolves collisions by probing the table according to a permutation of the integers $[1,m]$. The permutation used depends only on the key of the record in question. Thus for each key, the order in which the table is probed is a random permutation of all table locations. This method will equally likely use any of the $m!$ possible paths.

Uniform probing is a theoretical hashing model which has the advantage of being relatively simple to analyze. The following list summarizes some of the pertinent facts about this scheme:

$$Pr\{A_n' > k\} \;=\; \frac{n^{\underline{k}}}{m^{\underline{k}}}$$

where $n^{\underline{k}}$ denotes the descending factorial, that is, $n^{\underline{k}} = n(n-1)\cdots(n-k+1)$.

$$E[A_n] \;=\; C_n \;=\; \frac{m+1}{n}(H_{m+1} - H_{m-n+1}) \;\approx\; -\alpha^{-1}\ln(1-\alpha)$$

$$\begin{aligned}\sigma^2(A_n) \;&=\; \frac{2(m+1)}{m-n+2} - C_n(C_n+1) \\ &\approx\; \frac{2}{1-\alpha} + \alpha^{-1}\ln(1-\alpha) - \alpha^{-2}\ln^2(1-\alpha)\end{aligned}$$