Lecture 26
Hashing

November 15, 2021
Monday

INTRODUCTION

- Linear Search gives us: *O (n)*
- Binary Search gives us: *O (log n)*
- What about their best time
  - *O (1)*
- Can we maintain the *O (1)* for all three cases?
  - Best, Average, Worst ?

INTRODUCTION

What we mean by *O (1)* ?

Regardless of the number of elements being searched, the run time is always the same.

INTRODUCTION

What we mean by *Key ?*

Data we are looking in our data structure.

INTRODUCTION

Can we use a Key to represent the data?

A simplest example is the array, where key is the index and value is stored on that index..!

INTRODUCTION

Can you provide some examples of Key, Value pair data?

What about NIC number?
Your registration number?
Your Passport number?

HASH FUNCTION

What about Keys which are not in index form?

How we can save data then?

We need to find a function **h** that can transform a particular key **K**, be it a string, number, record, or the like, into an index in the table used for storing items of the same type as K. The function h is called a hash function.

HASH FUNCTION

If **h** transforms different **keys**

into different numbers, it is called a

perfect hash function.

HASH FUNCTION

To create a perfect hash function,

which is always the goal,

the table has to contain **at least the same number of positions as the number of elements being hashed.**

But the number of elements is

not always known ahead of time.

EXAMPLE

- Compiler keeps all variables used in a program in a symbol table.
- Real programs use only a fraction of the vast number of possible variable names, so a table size of 1,000 cells is usually adequate.
- Can we design a function **h**, which allows the compiler to immediately access the position associated with each variable?

EXAMPLE

- Consider adding all the letters fo the variables togather and the sum can be used as an index.
  - Table needs 3,782 cells
  - For a variable made out of 31 letters h (k) = 31 x 122 = 3, 782
- Even with this size, the function does not return unique value
  - h (“abc”) = h (“bca”)

HASH FUNCTION | COLLISION

When for different keys the function returns the same number, this is called **Collision.**

GOOD HASH FUNCTION

- The worth of a hash function depends on how well it avoids collisions.
- This can be achieved by making hash function more sophisticated.
  - But it should not be too sophisticated that the computation cost gets really big.
  - Less sophisticated methods may be faster.
- Should be fully deterministic
  - The hash values generated by hash functions should be valid bounds.
- Should distribute the values uniformly.
- Should generate different hash values for data very similar.

HASH FUNCTIONS

- The number of hash functions that can be used to assign positions to n items in a table of m positions (for $n \leq m$) is equal to m^n.
- However the number of perfect hash functions
  - m ! / (m - n) !
  - For example for 50 elements and a 100 cell array,
  - there are $100^{50} = 10^{100}$
    - Out of which only 10^{94} (one in a million)
    - Most of which are too unwieldy for practical applications.
    - And cannot be represented by a concise formula.

HASH FUNCTIONS | DIVISION

- A hash function must guarantee that the number it returns is a valid index to one of
- The simplest way is to use division modulo
  - TSize = sizeof(table)
  - h(K) = K mod TSize
- It is best if TSize is a prime number, or some prime > TSize can be used as well.
- However, nonprime divisors may work equally well as prime divisors provided they do not have prime factors less than 20

HASH FUNCTIONS | FOLDING

- The key is divided into several parts.
- These parts are combined or folded together and are often transformed in a certain way to create the target address.
- Two types of folding
  - Shift Folding
  - Boundary Folding

SHIFT FOLDING

- In Shift folding they are put underneath one another and then processed.
- Consider a CNIC Number 12345-1234567-1
- 12345, 1234567, 1
- Then these parts can be added.
- The resulting number 1,246,913 can be divided by modulo TSize.
- or in a table of 10000, first four digits can be used as address.
- Remember we can divide the number into any different parts,
    - 123, 45, 1234, 567, 1

BINARY FOLDING

- The key is seen as being written on a piece of paper that is folded on the borders between different parts of the key
- Consider Number 123456789
  - 123, 456, 789
- The first part 123 is taken in the same order, then the piece of the paper for 456 is folded underneath it so that 123 is aligned with 456
- When the folding continues, 789 is aligned with two previous parts.
  - 123 + 654 + 789 = 1, 566
-

FOLDING

- In both versions, the key is usually divided into even parts of some fixed size plus some remainder and then added.
- This process is simple and fast, especially when bit patterns are used instead of numerical values.
- A bit-oriented version of shift folding is obtained by applying the exclusive-or operation,

MID-SQUARE FUNCTION

- In the mid-square method, the key is squared and the middle or mid part of the result is used as the address.
- If the key is a string, it has to be preprocessed to produce a number by using, for instance, folding.
- The entire key participates in generating the address so that there is a better chance that differ-ent addresses are generated for different keys.

MID-SQUARE FUNCTION

- Consider key 3,121
  - $(3121)^2$ = 9, 740, 641 and for the 1,000 cell table
  - h(3121) = 406
  - Which is the middle part of 94**406**41
- It is more efficient to choose a power of 2 for the size of the table and extract the middle part of the bit representation of the square of a key

EXTRACTION

- Only part of the Key is used to compute the address.
- Consider the example 123-45-6789
  - This method might use 1234 OR 6789
  - Or Combination of first and second part
    - 1289
  - Or any other combination
- The common part of the key can be omitted in such case.
  - For example 20K-1234.

RADIX TRANSFORMATION

- The key is transformed into another number base.
  - K is expressed in numerical system using a different radix.
- Consider 345 in the decimal system.
  - Its base 9 value is 423.
  - This value can then be divided by modulo TSize and the resulting number is used as the address of the location to which K should be hashed.
- Collision, however, cannot be avoided.