Hashing

How to use hash tables to solve olympiad problems

What is hashing

- Hashing is used to allow elements of a set to be accessed directly

- Hashing is used when possible data sets are too big for arrays, but the actual data sets are quite small

When is hashing used?

- Hashing is used most often to encode a string so that it can be easily stored or processed.

- It is often necessary to keep track of whether strings exist, or some other fact about them. Hashing allows this data to be retrieved almost in O(1) time.

How does hashing work?

- Hashing follows a number of steps:
  - First, the data items are converted to a natural number
  - This number is then converted, using a hash function, to a different number which falls within the hash table
  - The original data is then stored in the hash table, where it can be quickly retrieved

Converting to natural numbers

- If numerical data is being hashed, it can easily be converted to natural numbers

- If text is being converted, it is easiest to use the ASCII values and compute a “value” for the string. This can be done in a number of ways:

- The string can be taken as a number base 256, which can be converted to decimal

- A simpler method is to multiply letters by prime numbers and add them together.
  - Eg “hello” - take the value of h, add to it the value of the e multiplied by a prime, eg 29, then add the value of the l, multiplied by 29 and 31, etc.

- Primes are used to reduce patterns

Using a hash function

- In the previous step, each element was given a key, which had a reasonalbe chance of being unique.
- These keys are too big to be used directly in an array, so they are hashed first
- The simplest hash function involves modding the key by the table size.

A hash function

- If a given element has a key *k*, and is being inserted into a hash table of size *m*, the following is a simple and reasonably effective hash function:

- $h(k) = k \bmod m$

- If this hash function is used, m should be prime to increase the spread of data

Inserting data into a hash table

- Before data can be added, the table must be created. The size is reasonably important

- Unless using linked lists, the table should be 1.5 to 2 times the size of the data being stored in it, and its size should be a prime (or space is wasted)

Inserting data

- Once the key has been hashed, it gives some value *h(k)*. The original data can then be inserted into the array with this as index

- If this space is full, keep checking the next space till an open space is found. Wrapping around may be necessary

Finding an element in the table

- To find an element, first compute its key and then pass this through the hash table.

- Go to this entry in the table. If it doesn’t contain the correct entry, check the next one. Repeat this till the entry is found or and empty space occurs

Deleting an item from a hash table

- To delete an item, first locate it using the search procedure
- The block should be marked with some code or flag which indicates it is “deleted”, but not in its original state

Using linked lists in hash tables

- Instead of inserting items below the index in the table, they can form a linked list starting at the index

- When using this data structure, which needs pointers, the table does not need to be larger than the number of entries

Problems using hashing

- A very simple problem which can be solved using hashing is the problem of counting the frequencies of words in a piece of text

- These are then stored in a hash table, with the word itself as the key

Problems using hashing

- Decrypting a piece of text with a simple code where a letter is replaced by a letter *k* further in the alphabet

- For various values of *k*, the text can be checked to find which gives the best match.

Problems using hashing

- Finding which words are synonyms given a list of pairs of words with the same meaning

- This becomes graph theory, and is solved with a flood fill algorithm

Problems using Hashing

- Although these all relate to text, hashing can be used to replace any akward data with a direct access table

- This can be used for polygons, long sequences of numbers, etc

Conclusion

- Hashing is used to allow data like text to be stored so that it can be accessed in almost O(1) time.

- It is often only part of the solution, and is used to simplify a problem so that an method like dynamic programming or a graph theory algorithm can be used