

ACOL106: Data Structures and Algorithms

Ragesh Jaiswal, IIT Delhi

Data Structures: Hashing

- We have seen data structures for storing and accessing *entries* (key-value pairs) such that the running time for each of the operations is:
 - Search: $O(\log n)$
 - Insert: $O(\log n)$
 - Delete: $O(\log n)$
- Question: Can you design a data structure with the following running time?
 - Search: $O(1)$
 - Insert: $O(1)$
 - Delete: $O(1)$

- Question: Can you design a data structure with the following running time?
 - Search: $O(1)$
 - Insert: $O(1)$
 - Delete: $O(1)$
- Suppose for the sake of simplicity that all keys are positive integers.
- Main idea: Use an array indexed by the keys and store the entry with key i at $A[i]$.

- Question: Can you design a data structure with the following running time?
 - Search: $O(1)$
 - Insert: $O(1)$
 - Delete: $O(1)$
- Suppose for the sake of simplicity, all keys are positive integers.
- Main idea: Use an array indexed by the keys and store the entry with key i at $A[i]$.
- Question: What is the main issue with this idea?

- Question: Can you design a data structure with the following running time?
 - Search: $O(1)$
 - Insert: $O(1)$
 - Delete: $O(1)$
- Suppose for the sake of simplicity, all keys are positive integers.
- Main idea: Use an array indexed by the keys and store the entry with key i at $A[i]$.
- Question: What is the main issue with this idea?
 - Wastage of space.

Data Structures

Hashing

- Question: Can you design a data structure with the following running time?
 - Search: $O(1)$
 - Insert: $O(1)$
 - Delete: $O(1)$
- Suppose for the sake of simplicity, all keys are positive integers.
- Main idea: Use an array indexed by the keys and store the entry with key i at $A[i]$.
- Question: What is the main issue with this idea?
 - Wastage of space.
- Question: How do we fix this issue?

Data Structures

Hashing

- Question: Can you design a data structure with the following running time?
 - Search: $O(1)$
 - Insert: $O(1)$
 - Delete: $O(1)$
- Suppose for the sake of simplicity, all keys are positive integers.
- Main idea: Use an array indexed by the keys and store the entry with key i at $A[i]$.
- Question: What is the main issue with this idea?
 - Wastage of space.
- Question: How do we fix this issue?
 - Use array $A[0 \dots N - 1]$ and store an entry with key k at $A[h(k)]$, where $h : K \rightarrow \{0, \dots, N - 1\}$, where K denote the space of keys.

Data Structures

Hashing

- Question: Can you design a data structure with the following running time?
 - Search: $O(1)$
 - Insert: $O(1)$
 - Delete: $O(1)$
- Suppose for the sake of simplicity, all keys are positive integers.
- Main idea: Use an array indexed by the keys and store the entry with key i at $A[i]$.
- Question: What is the main issue with this idea?
 - Wastage of space.
- Question: How do we fix this issue?
 - Use array $A[0 \dots N - 1]$ and store an entry with key k at $A[h(k)]$, where $h : K \rightarrow \{0, \dots, N - 1\}$, where K denote the space of keys.
- Question: What is the new issue raised by the above idea?

Data Structures

Hashing

- Question: Can you design a data structure with the following running time?
 - Search: $O(1)$
 - Insert: $O(1)$
 - Delete: $O(1)$
- Suppose for the sake of simplicity, all keys are positive integers.
- Main idea: Use an array indexed by the keys and store the entry with key i at $A[i]$.
- Question: What is the main issue with this idea?
 - Wastage of space.
- Question: How do we fix this issue?
 - Use array $A[0 \dots N - 1]$ and store an entry with key k at $A[h(k)]$, where $h : K \rightarrow \{0, \dots, N - 1\}$, where K denote the space of keys.
- Question: What is the new issue raised by the above idea?
 - Collision: There may exist keys $i \neq j$ such that $h(i) = h(j)$.

Data Structures

Hashing

- Question: Can you design a data structure with the following running time?
 - Search: $O(1)$
 - Insert: $O(1)$
 - Delete: $O(1)$
- Suppose for the sake of simplicity, all keys are positive integers.
- Main idea: Use an array indexed by the keys and store the entry with key i at $A[i]$.
- Question: What is the main issue with this idea?
 - Wastage of space.
- Question: How do we fix this issue?
 - Use array $A[0 \dots N - 1]$ and store an entry with key k at $A[h(k)]$, where $h : K \rightarrow \{0, \dots, N - 1\}$, where K denote the space of keys.
- Question: What is the new issue raised by the above idea?
 - Collision: There may exist keys $i \neq j$ such that $h(i) = h(j)$.
- Question: How do we avoid collisions (as much as possible)?
- Question: How do we resolve collisions?

Data Structures

Hashing

- Main idea: Use array $A[0 \dots N - 1]$ and store an entry with key k at $A[h(k)]$, where $h : K \rightarrow \{0, \dots, N - 1\}$, where K denote the space of keys.
- Question 1: How do we avoid collisions (as much as possible)?
- Question 2: How do we resolve collisions?

Data Structures

Hashing → Avoiding Collision

- Question 1: How do we avoid collisions (as much as possible)?
- The nature of keys of entries may be varied depending on the context (*it may not always be a positive integer as we assumed*):
 - In case of school records, the key may be the identification number of students.
 - In case of a file system, it may be the file identifier.
 - In case of photograph storage, it may be the photos themselves.

End