Here comes text in 11pt Times New Roman font... please check spelling and grammar.

Last time: Hashing
Today: Hashing + Heaps

Hashing with open addressing!

\[ S, |S| \leq m, |S| \ll |U| \]

\[ \text{Rehash } y \]

Let’s have \( m \) hash functions: \( h_1, h_2, h_3 \ldots h_m \)

Such that for \( \forall i \in \{1, 2 \ldots m\} \)
\( h_i : U \rightarrow \{1, 2 \ldots m\} \)

and that
\( \forall x \ ( h_1(x), h_2(x) \ldots h_m(m) ) \) is a permutation of \( (1, 2, 3 \ldots m) \)

Example:
We assume \( h_1(x) = x \mod 7 + 1 \)
\( h_2(x) = (2x) \mod 7 + 1 \)

\( h_1(15) = 2 \)
\( h_1(8) = 2 \text{----------collision!} \)
Then we use \( h_2, h_2(8) = 3 \)

Assume: each permutation \( (1, 2 \ldots m) \) is assigned to \( \frac{|U|}{m!} \) elements of \( U \)

Cost analysis: execute \( n \) operations and ask what is the expected cost of \( (n+1)^{\text{th}} \) operation

\( P_i \) is the probability that we will rehash at least \( i \) times

\( P_0 = 1 \)
\( P_1 = \frac{n}{m} \)

\[ \text{Lecture Notes for I500} \]

\[ \text{Fall 2009} \]

\[ \text{Fundamental Computer Concepts of Informatics} \]

\[ \text{by Xiaoqian Zhang (Instructor: Predrag Radivojac)} \]
\[
\begin{align*}
\mathbf{P}_2 &= \frac{n \cdot n - 1}{m \cdot m - 1} \\
\vdots \\
\mathbf{P}_i &= \frac{n \cdot n - 1}{m \cdot m - 1} \cdots \frac{n - i + 1}{m - i + 1} \\
\end{align*}
\]

Probability to rehash exactly \( i \) times: \( P_i - P_{i+1} \)

\[
C = (P_0 - P_1)*1 + (P_1 - P_2)*2 + \ldots + (P_n - P_{n+1}) (n + 1)
\]

\[
C \text{ is the cost of (n+1)th operation!}
\]

\[
= P_0 + P_1 + P_2 \ldots + P_n - (P_{n+1})*(n+1) \quad \text{we know that}= 1; \quad P_{n+1} = 0
\]

\[
= 1 + P_1 + P_2 \ldots + P_n
\]

Theorem 1: Under previous assumptions, the expected cost of \((n+1)^{th}\) operation is at most

\[
\frac{m+1}{m-n+1} \approx \frac{1}{1 - \alpha}
\]

\( n \) = \( n/m \)

We prove the Theorem:

Base case: \( n = 0 \); \( C = \frac{m+1}{m+1} = 1 ; P_0 = 1 \)

Recurrent step: Assume it holds for \( m-1, n-1 \), that’s \( 1 + \frac{n-1}{m-1} + \ldots + \frac{n-1}{m-1} \cdots \frac{n-n+1}{m-n+1} = \frac{(m-1)+1}{(m-1)-(n-1)+1} \)

The goal is to prove

\[
1 + P_1 + P_2 \ldots + P_n = \frac{m+1}{m-n+1}
\]

\[
= 1 + \frac{n}{m} + \frac{n}{m} \cdot \frac{n-1}{m-1} + \ldots + \frac{n}{m} \cdot \frac{n-1}{m-1} \cdots \frac{n-n+1}{m-n+1}
\]

\[
= 1 + \frac{n}{m} \cdot (1 + \frac{n-1}{m-1} + \ldots + \frac{n-1}{m-1} \cdots \frac{n-n+1}{m-n+1}) \quad \text{(By IH)}
\]

\[
= 1 + \frac{n}{m} \cdot \frac{(m-1)+1}{(m-1)-(n-1)+1}
\]

\[
= 1 + \frac{n}{m-n+1} = \frac{m+1}{m-n+1}
\]

Summary: Cost of the \( n^{th} \) operation: \( = O\left(\frac{1}{1 - \alpha}\right) \) for hashing with open addressing.

\( O(1+\alpha) \) for chaining
If $\alpha < \frac{1}{2}$ we can use open addressing to save memories; if $\alpha > \frac{1}{2}$ we better use chaining