Subject : ....

* hash table of K entries
after n keys

if $\begin{cases}\frac{n}{K} \text{ large} \Rightarrow \text{ hash won't work} \\ \text{load factor} \\ \\ \text{hash non-uniform} \Rightarrow \frac{n}{K_{eff}} \text{ large}\end{cases}$
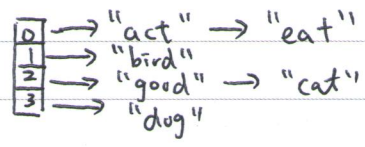
* idea: increase K when $\frac{n}{K}$ large

* naive
 ① set $K^{new} = 2K$
 ② change $h(key)$ to range $\{0, \cdots, 2K-1\}$
 ③ rebuild w/ $O(n)$ if insert is $O(1)$
 • cannot do often $(\frac{n}{k} > \theta)$
 • long waiting

* lazy approach
 ① set $K^{new} = 2K$ (use one more bit of $h(\cdot)$)
 ② change $h(key)$
 ③ rebuild only the overflow entry $O(k) + O(\frac{n}{k})$
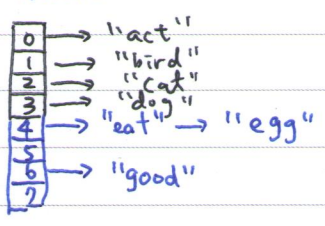
 e.g. hashing w/ chaining of length 2
   $h(key) = (key[0] - 'a') \% K$

   insert "egg"

naive

lazy (directory extension)

Double A