

Design and Analysis of Algorithms I

#### Data Structures

Universal Hash Functions: Definition and Example

### Overview of Universal Hashing

Next: details on randomized solution (in 3 parts).

Part I: proposed definition of a "good random hash functions". ("universal family of hosh functions")

Part II. Concrete example of simple + practical such functions

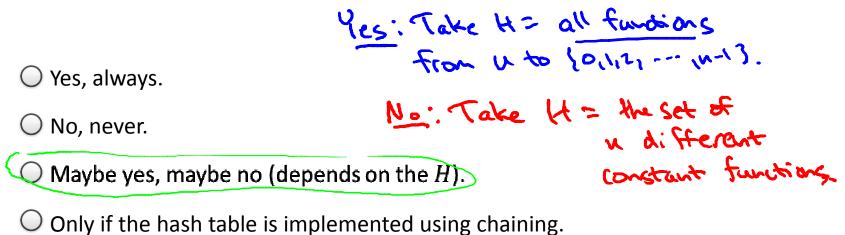
Part III: 'yestification et definition: "good functions" lead to "good performance".

#### Universal Hash Functions

Definition: Let H be a set of hash functions from U to {0,1,2,..., N-13. His universal if and only if: for all xiy E a (with x+y) ( of bodiets) Pr [x,y collide] = [n] when h is chosen uniformly at roundon from H. (i.e., collision probability as small as with "gold standard" of persector random hashing)

Tim Roughgarden

Consider a hash function family H, where each hash function of H maps elements from a universe U to one of n buckets. Suppose H has the following property: for every bucket i and key k, a 1/n fraction of the hash functions in H map k to i. Is H universal?



# Example: Hashing IP Addresses

#### A Universal Hash Function

Ortine: H = { ha | e 101/21-14-3}. (a,x,+a,x,+a,x,+a,x,+a,x,+)

Noon

Theorem: This family is universal.

# Proof (Part I)

Consider district IP addresses (x1, x2, x3, x4), (x1, x2, y3, y4).

Assure: x4 + y4. Question: collision probability?

(i.e., Prob [halx1, 1, x4) = haly1, 1, y4)

Note: collision (=>  $a_1x_1+a_2x_2+a_3x_3+a_4x_4 = a_1y_1+a_2y_2+a_3y_3+a_4y_4$ (mod n)  $a_1(y_4-y_4)$  mod  $a_1=\frac{3}{3}$   $a_1(y_1-x_1)$  mod  $a_2$ 

Next: Condition on random choices of a, 1 az, az, az.

#### Proof (Part II)

The Story So Far: with a vazing Fitch arbitrarily, how Choices & an satisfy (autry-yn) mat in Key Claim: left-hand side equally likely to be any & 80,1,2, ..., 11-13. beason: xy+yy (xy-yy+0 mod n) (Caldendam: Proof" by example: n=7, xu-Yu=2 or 3 mod n Prob[ha(x)=ha(y)]=+ COED! Tim Roughgarden