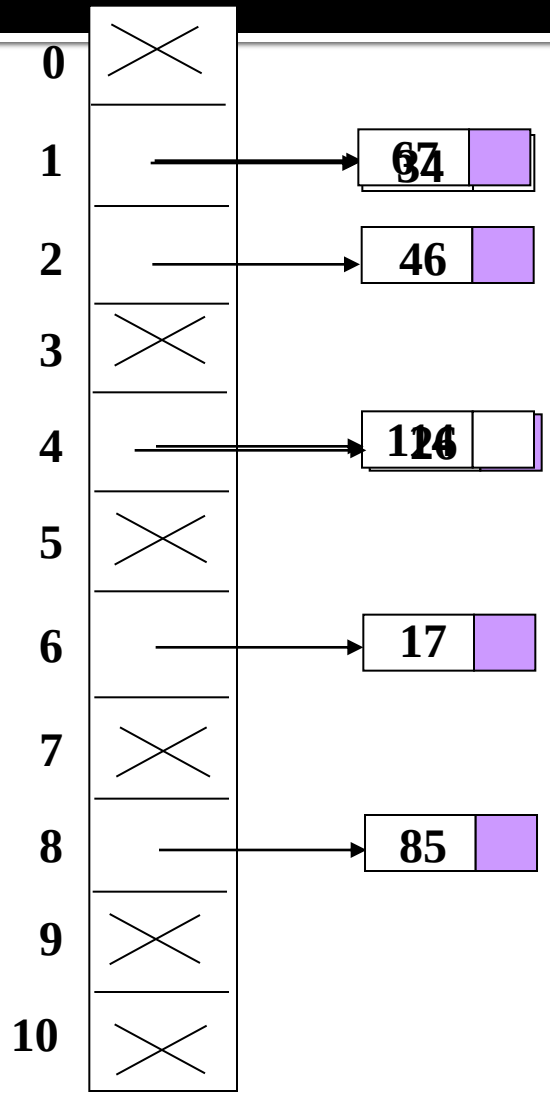
Lab 10: Hash Tables with Chaining

EPL231 – Data Structures and Algorithms

Διαχείριση συγκρούσεων με αλυσίδωση

- Αφού περισσότερα από ένα κλειδιά μπορούν να πάρουν την ίδια τιμή από τη συνάρτηση κατακερματισμού, μπορούμε να θεωρήσουμε ότι κάθε θέση του πίνακα 'δείχνει' σε μια ευθύγραμμη απλά συνδεδεμένη λίστα.
- Για κάθε i, στη θέση H[i] του πίνακα βρίσκουμε λίστα που περιέχει όλα τα κλειδιά που απεικονίζονται από τη συνάρτηση h στη θέση αυτή.
- Για να βρούμε κάποιο κλειδί k, πρέπει να ψάξουμε στη λίστα που δείχνεται στη θέση H[h(k)].
- Εισαγωγές και εξαγωγές στοιχείων μπορούν να γίνουν εύκολα με βάση τις διαδικασίες συνδεδεμένων λιστών.

Παράδειγμα



hsize = 11

Εισαγωγή:

Συναρτήσεις για υλοποίηση

- *Πίνακας κατακερματισμού σταθερού μεγέθους (TABLE_SIZE):*
  - `void destroyTable(HashTable * table);`
  - `void insertEntry(HashTable *table, Student *student);`
  - `void deleteEntry(HashTable *table, const char *studentID);`
  - `HashEntry *findEntry(HashTable *table, const char *studentID);`
- *Εισαγωγή 10000 φοιτητών με τυχαίο ID*
- *Τύπωμα του πίνακα κατακερματισμού*

Something COOL!!!

```
/*************************************************************************
* *
* ------------------------------ hashpjw ------------------------------ *
* *
*************************************************************************/
int hashpjw(const void *key) {
    const char *ptr;
    int val;
    val = 0;
    ptr = key;

    while (*ptr != '\0') {
          int tmp;
          val = (val << 4) + (*ptr);
          if (tmp = (val & 0xf0000000)) {
                    val = val ^ (tmp >> 24);
                    val = val ^ tmp;
          }
          ptr++;
    }
    return val % PRIME_TBLSIZ;
}
```

The End