

EECS2011 Fundamentals of Data Structures
(Winter 2022)

Q&A - Syllabus

Tuesday, January 11

EECS
[]

→ Jan 31st

↳ online

EECS 2011 : assumes

EECS 2020 / EECS 1021

↓
proficiency in Java.

↳ recursion is important!

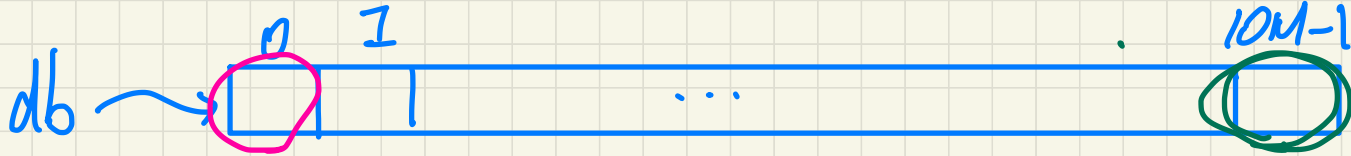
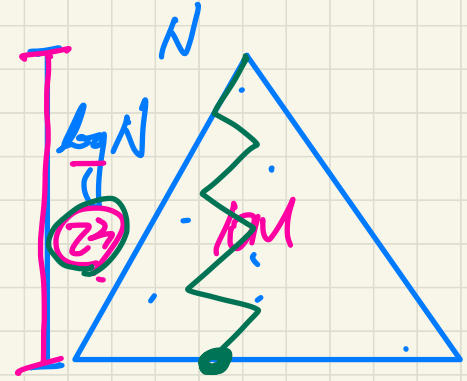
youtube.

obama bubble sort

A Searching Problem

```
ResidentRecord find(int sin) {  
    for(int i = 0; i < database.length; i++) {  
        if(database[i].sin == sin) {  
            return database[i];  
        }  
    }  
}
```

balanced BST



How efficient? (# of iterations)

→ best case: I

→ worst case: $10M$

Program Optimization Problem

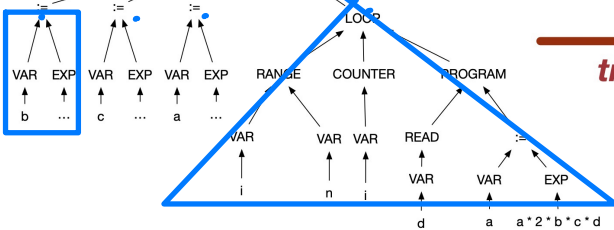
```
b := ... ; c := ... ; a := ...  
across i |...| n is i  
loop  
  read d  
  a := a * 2 * b * c * d  
end
```

```
b := ... ; c := ... ; a := ...  
temp := 2 * b * c  
across i |...| n is i  
loop  
  read d  
  a := a * d * temp  
end
```

optimized

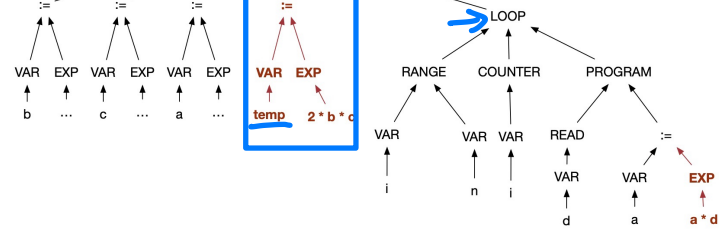
parsed

PROGRAM



transformed

PROGRAM



pretty-printed

Complex. EEIS 4302. EEIS 2001. CFG.



Program Translation Problem

```
class Account {
  attributes
  owner: Traveller . account
  balance: int
}
```

```
class Traveller {
  attributes
  name: string
  reglist: set(Hotel . registered)[*]
}
```

```
class Hotel {
  attributes
  name: string
  registered: set(Traveller . reglist)[*]
  methods
  register {
    t? : extent(Traveller)
    & t? /: registered
    ==>
    registered := registered \ {t?}
    || t?.reglist := t?.reglist \ {this}
  }
}
```

translated

```
CREATE TABLE 'Account'({
  'oid' INTEGER AUTO_INCREMENT, 'balance' INTEGER,
  PRIMARY KEY ('oid'));
CREATE TABLE 'Traveller'({
  'oid' INTEGER AUTO_INCREMENT, 'name' CHAR(30),
  PRIMARY KEY ('oid'));
CREATE TABLE 'Hotel'({
  'oid' INTEGER AUTO_INCREMENT, 'name' CHAR(30),
  PRIMARY KEY ('oid'));
CREATE TABLE 'Account_owner_Traveller_account'({
  'oid' INTEGER AUTO_INCREMENT, 'owner' INTEGER, 'account' INTEGER,
  PRIMARY KEY ('oid'));
CREATE TABLE 'Traveller_reglist_Hotel_registered'({
  'oid' INTEGER AUTO_INCREMENT, 'reglist' INTEGER, 'registered' INTEGER,
  PRIMARY KEY ('oid'));
```

parsed

Abstract Syntax Tree of
Source Object-Oriented Program

transformed

Abstract Syntax Tree of
Target Relational DB Queries

pretty-printed