DYNAMIC DATA STRUCTURES, A WEB BASED TOOL FOR TEACHING LINKED LISTS AND BINARY TREES

Dynamic Data Structures

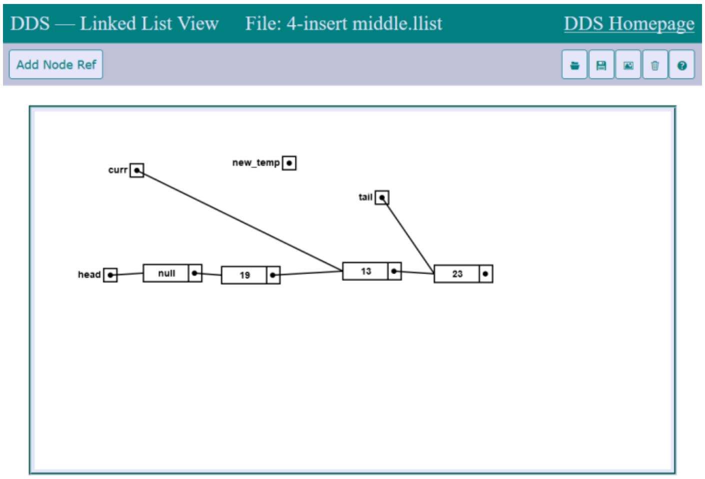
Robert A. Ravenscroft, Jr.
Rhode Island College

Motivation and Goals

- Linked lists and binary trees are dynamic
- White boards are static
- Dynamic, web based classroom teaching tool
  - **No hand drawing**
  - Model and manipulate data structures
  - Syntactically correct, consistent with Java
  - Easy to use
- **Not an algorithm visualization system!**

Dynamic Data Structures

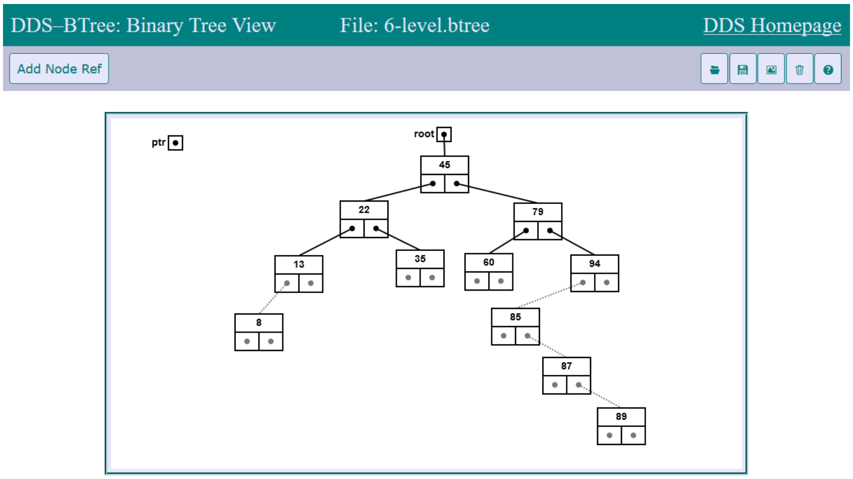
Linked Lists



DDS — Linked List View File: 4-insert middle.llist DDS Homepage

Dynamic Data Structures

Binary Trees



DDS–BTree: Binary Tree View File: 6-level.btree DDS Homepage

DDS Availability

- dsviewer.org

- dsviewer.org/dds-llist
- dsviewer.org/dds-btree
- dsviewer.org/dds-homepage

Node Object for a List

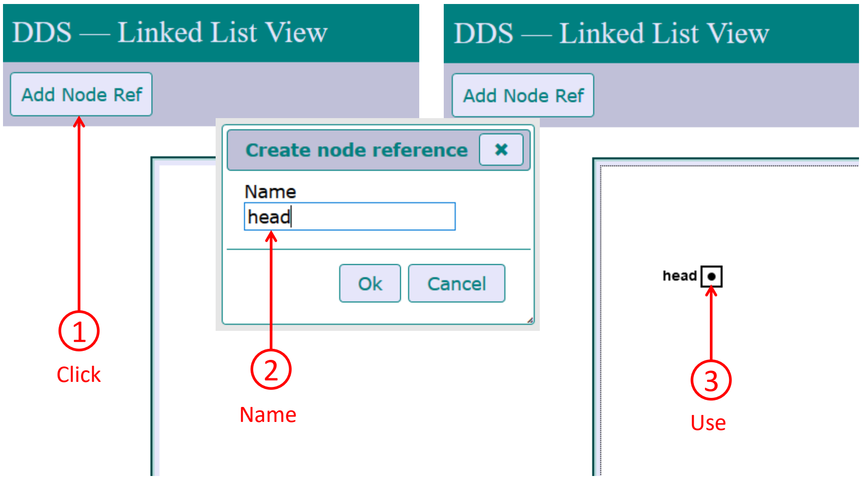
```
Class Node{
    String payload;
    Node next;
};

Node head, tail;
```

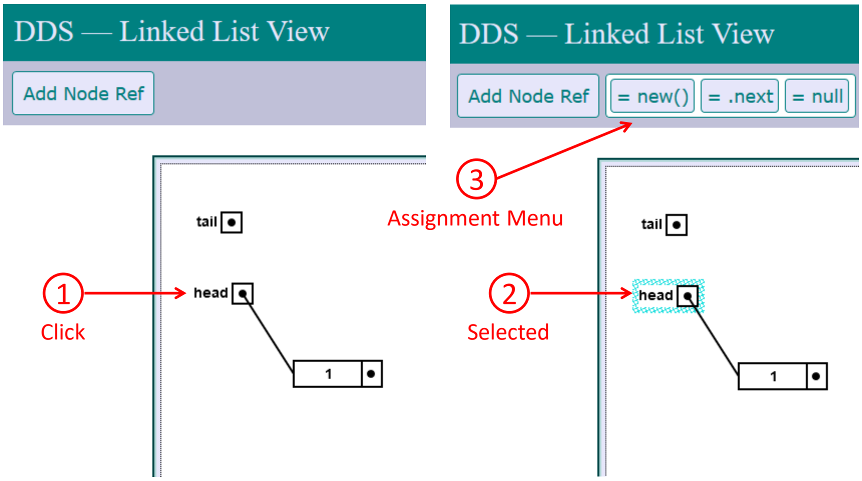
Five Easy Steps to Use DDS

1. Add a node reference variable
2. Select a node or reference variable, enable the assignment menu
3. Assignment menu: allocate a new node, assign null, or goto next
4. Reference assignment
5. Drag a node or reference variable

1. Create a Node Reference



2. Select Node/Reference Variable

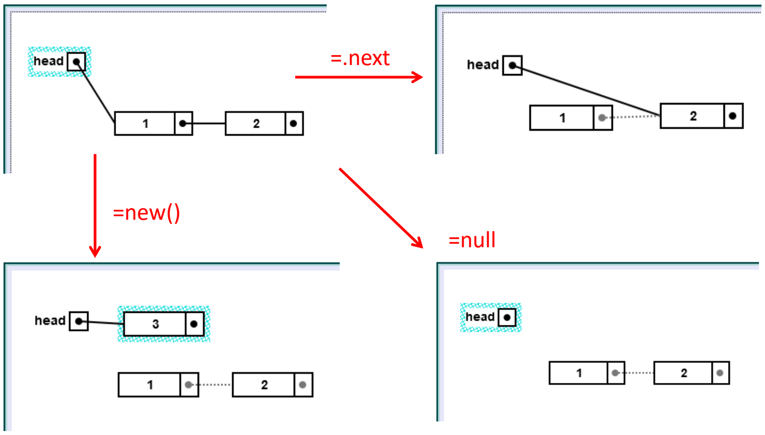


3 Assignment Menu

1 Click

2 Selected

3. Assignment Menu



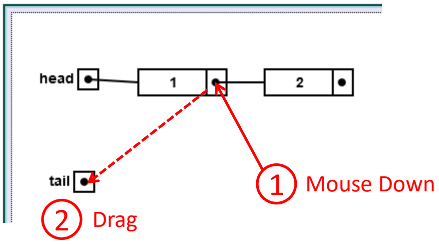
=.next

=new()

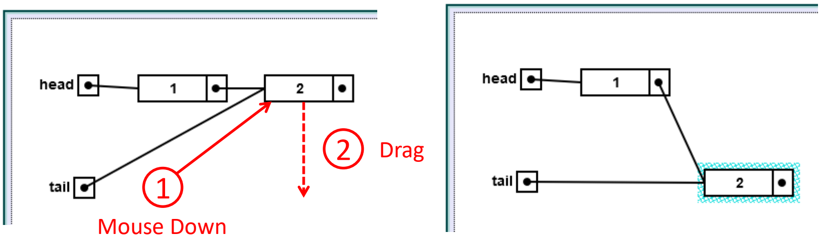
=null

4. Reference Assignment

tail = head.next;

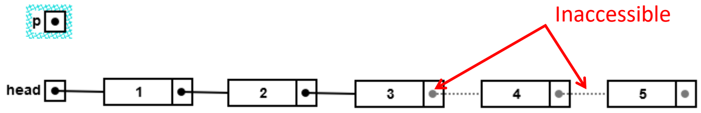


5. Drag Node/Reference Variable

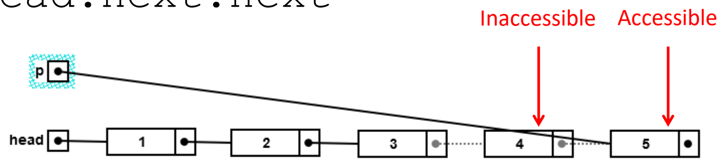


Accessible Nodes

- Access at most two next references

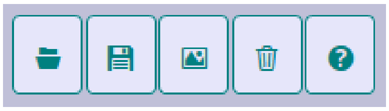


- `head.next.next`

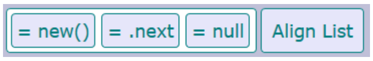


Other Options

- Load
- Download
- Image
- Garbage
- Help



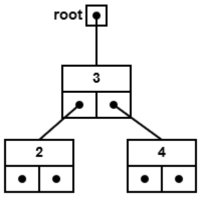
- Align



Binary Trees

```
Class Node{
    String payload;
    Node left;
    Node right;
}

Node root;
```



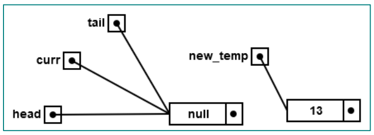
Classroom Experience

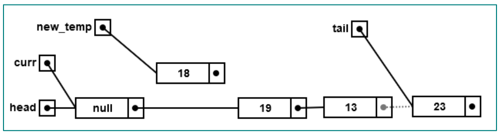
- Algorithms and Data Structures
  - DDS-LList: List, Stack, Queue ADTs
  - DDS-BTree: Binary Search Tree, Dictionary ADT
- Analysis of Algorithms
  - Linked lists and balanced binary trees
  - Model traversal and searching
  - Motivate run time analysis

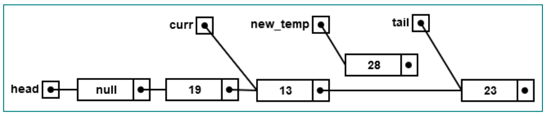
Classroom Usage

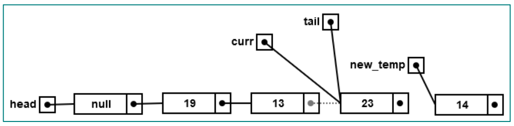
- Introduce concepts of linked lists and binary trees
- Motivate, with student input, how (and how not to) implement an algorithm
- Trace algorithm behavior

Insert into List ADT

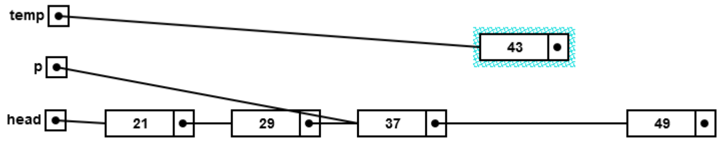




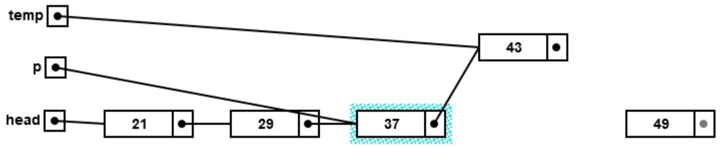




How Not to Insert a Node



```
p.next = temp;   temp.next = ?
```



Classroom Benefits

- Pre-built scenarios
- Eliminate the draw-erase-redraw cycle
- Concentrate on concepts and not on data structure maintenance
- Eliminate concerns with an incorrect or corrupt model
- Quickly reload scenarios
- Post data files on course web site

Binary Trees

- Motivate traversal algorithms
- No recursion feature, unable to model or implement traversal algorithms
- Motivate and implement algorithms to descend a path to a leaf node
  - Search
  - Insert
  - Delete

Student Impact

- No evaluation, meant to aid instructor
- Negative
  - Students still had to draw-erase-redraw
  - Need to provide support material on web site
- Unexpected benefit
  - Easier to answer student questions
  - Easy to model a scenario for question
  - Return to scenario under discussion

Needed Necessary Features

- Recursion
  - Keep it simple
  - Allow a reference variable to be pushed/popped?
  - Activation records add too much complexity?
- Tabbed interface
- Undo
- Capture snapshots or record actions for playback

Additional Useful Features

- Copy and paste
- Multiple selection and drag
- Data structure layout commands
- Integer or key-value data payloads
- Highlight tool
- Node coloring
- Command palette
- More on dsviewer.org/dds/homepage

Possible Student Usage of DDS

- Limited options for active learning
- Need recording or snapshot feature to submit steps used in an assignment solution
- Block language has been prototyped
  - Algorithm visualization
  - Algorithm development and testing
  - Prototype available on dsviewer.org/dds/homepage

Conclusion

- Met objectives (no hand drawing!)
- Focus on concepts, not on drawing models
- Reload examples with a clean model
- Easy to post class examples to web/lms
- Responding to student questions (unexpected)

- Future work: continue development of modeling tools for data structures and ADTs

DDS is Seeking …

- Users
- Suggestions for features and improvements
- Bug reports
- Collaboration on classroom usage
- Ideas for student usage
- Evaluation of student usage

- Let us know about your DDS experiences

Thank you!

Questions?