FINAL PRACTICE

Final exam: Online via Gradescope

Time: Monday June 6, 4p - 6p

Read the instructions for the exam carefully:
https://ucsb-cs24.github.io/s22/exam/e02/

- The runtime complexity of an algorithm is $T(n) = 5n^2logn + n + 1000$
- Show that $T(n) = \theta(n^2logn)$ using the definition of Big-Theta

- Assume **dataA** is some data structure and the input vector **v** has N key
- Describe algoX in a sentence

```
void algoX(vector<int>& v)
{
        dataA ds;
        for(auto& elem: v)
            ds.insert(elem);
        for(auto& elem: v){
            elem = ds.min();
            ds.delete(elem);
        }
}
```

- Assume **dataA** is some data structure and the input vector to algoX has N numbers
- Given: running time of operations for dataA, where M is the number of keys stored in dataA
  - insert: O(log M)
  - min: O(1)
  - delete: O(log M)

```
void algoX(vector<int>& v)
{
        dataA ds;
        for(auto& elem: v)
            ds.insert(elem);
        for(auto& elem: v){
            elem = ds.min();
            ds.delete(elem);
        }
}
```

What is the Big-O running time of algoX?

A. $O(N^2)$
B. O(N logN)
C. O(N)
D. O(log N)
E. Not enough information to compute

Big Four and the Rule of Three

Implement the copy constructor for a BST

Data structure Comparison

	Insert	Search	Min	Max	Delete min	Delete max	Delete (any)
Sorted array							
Unsorted array							
Sorted linked list (assume access to both head and tail)							
Unsorted linked list							
Stack							
Queue							
BST (unbalanced)							
BST (balanced)							
Min Heap							
Max Heap							

Data structure Comparison

	Insert	Search	Min	Max	Delete min	Delete max	Delete (any)
Sorted array	O(N)	O(logN)	O(1)	O(1)	O(N) if ascending order, else O(1)	O(1) if ascending, else O(N)	O(logN) to find, O(N) to delete
Unsorted array	O(1)	O(N)	O(N)	O(N)	O(N)	O(N)	O(N)
Sorted linked list (assume access to both head and tail)	O(N)	O(N)	O(1)	O(1)	O(1)	O(1)	O(N) to find, O(1) to delete
Unsorted linked list	O(1)	O(N)	O(N)	O(N)	O(N)	O(N)	O(N) to find, O(1) to delete
Stack	O(1) - only insert to top	Not supported	Not supported	Not supported	Not supported	Not supported	O(1) - Only the element on top of the stack
Queue	O(1) - only to the rear of the queue	Not supported	Not supported	Not supported	Not supported	Not supported	O(1) - only the element at the front of the queue
BST (unbalanced)	O(N)	O(N)	O(N)	O(N)	O(N)	O(N)	O(N)
BST (balanced)	O(logN)	O(logN)	O(logN)	O(logN)	O(logN)	O(logN)	O(logN)
Min Heap	O(logN)	Not supported	O(1)	Not supported	O(logN)	Not supported	O(logN)
Max Heap	O(logN)	Not supported	Not supported	O(1)	Not supported	O(logN)	O(logN)