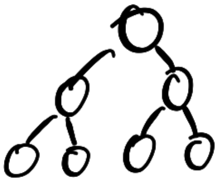


log n

CSE214 COMPUTER SCIENCE II
FINAL EXAM PRACTICE QUESTIONS

USE THE FOLLOWING INFORMATION TO ANSWER PROBLEMS 1.1-1.4:

Consider the following four operations on a data structure containing *n* data values.

O(n) A. Finding the maximum value in a singly linked list of *n* `IntNode` nodes. Traversal LL

O(n^2) B. Finding the maximum value in an array of *n* `int` values by sorting it first using insertion sort.

O(logn) C. Finding the maximum value in a full binary search tree of *n* `BTNode` nodes.

O(1) D. Finding the maximum value in a standard heap of *n* data values.

1.1 The worst-case order of complexity is O(1) for which of these operations?
(a) A (b) B (c) C [✓ (d) D] (e) none of these answers

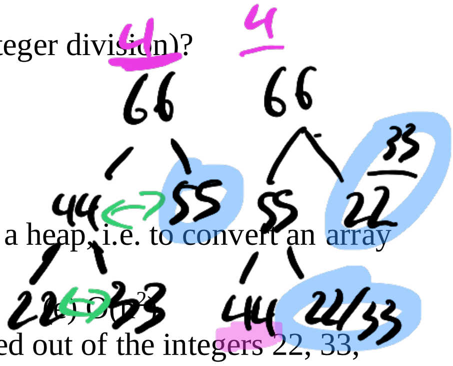
1.2 The worst-case order of complexity is O(log n) for which of these operations?
(a) A (b) B [✓ (c) C] (d) D (e) none of these answers

1.3 The worst-case order of complexity is O(n) for which of these operations?
[✓ (a) A] (b) B (c) C (d) D (e) none of these answers

1.4 The worst-case order of complexity is O(n log n) for which of these operations?
(a) A (b) B (c) C (d) D [✓ (e) none of these answers]

1.5 Which of the following postfix expressions evaluates to 35 (assuming integer division)?

(a) `5 4 3 2 1 + * - /` (b) `5 4 3 2 1 * - / +`
[✓ (c) `5 4 3 2 1 - / + *`] (d) `5 4 3 2 1 / + * -`
(e) none of these answers



1.9 What is the order of complexity for the most efficient algorithm to make a heap, i.e. to convert an array into a heap?
(a) O(1) (b) O(log n) [✓ (c) O(n)] (d) O(n * log n) (e) O(n^2)

1.10 How many different heaps (with the maximum at the root) can be formed out of the integers 22, 33, 44, 55, and 66?
(a) 5 (b) 6 (c) 7 [✓ (d) 8] (e) none of these answers

1.11 Which of following methods are tail-recursive?

```
public void A(int n) {
  if (n == 0) return;
  else {
     if (n % 2 == 0){
          System.out.println(n+"*");
           A(n-2);
     }
      else {
          System.out.println(n+"/");
          A(n-1);
    }
  }
```

```
public void B(int n) {
   if (n == 0) return;
   else {
      System.out.println(n);
       B(n-1);
   }
}
public void C(int n) {
   if ( n == 0) return;
   int[] t = {3,2,7};
   for (int i=0;i<3;i++) {
     System.out.println(t[i]);
     C(n-1);
```

```
    }                                              }
                                               }
```

(a) C (b) B and C [✓ (c) A and B] (d) all of the above (e) none of the above

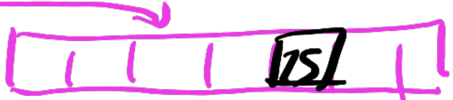
1.12 Which of the following trees have all of their leaves at the same level?

I. Red-black Tree ✗
II. B-Tree ✓
III. Complete binary tree ✗
IV. Full binary tree ✓

(a) Only II [✓ (b) II and IV] (c) II, III and IV (d) all of the above (e) none of the above

1.13 Consider the following double-hashing function for a hash table of size 100:

```
75   H1(k)  = k mod 100
     H2(k)  = 2 + (k mod 52)
```



For `k` = 75, how many elements of the hash table are examined in the worst case before an empty slot is found for `k`?

(a) 100 (b) 52 (c) 50 [✓ (d) 4] $\frac{100}{25}$ (e) 2

1.14 Which of the following is the best replacement for `H2(k)` given in problem 1.13 for `k` =75.

```
(a) H2(k) = 1 + (k mod 52)            (b) H2(k) =  5 + (k mod 52)
(c) H2(k) = 4 + (k mod 52)            (d) H2(k) = 27 + (k mod 52)
(e) none of these answers
```

[✓ (c)]

Use the following hash table to answer questions 2.1-2.3. The hash table stores integer keys using a hash function `h(k) = k mod 17`. All keys were inserted without collisions.

INDEX	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
KEY		69		88	4				59	94	27			47	31		16
HAS_BEEN_USED	F	T	F	T	T	F	F	F	T	T	T	F	F	T	T	F	T

2.1 At what position will the key 60 be stored in the hash table using h(k) above if linear probing is used to resolve collisions?

60 mod 17 = 9

9 - FULL 10 - FULL 11 - Good

(a) 2 (b) 5 [✓ (c) 11] (d) 15 (e) none of these answers

2.2 At what position will the key 60 be stored in the original hash table if double hashing is used to resolve collisions, assuming $h_1(k) = h(k)$ and $h_2(k) = 2 + (k \bmod 11)$?

$h_1 = 9$
$h_2 = 7 \to 16$
$\underline{16} + 7 \to 6$

(a) 0 [✓ (b) 6] (c) 11 (d) 12 (e) none of these answers

2.3 What is the load factor of the original hash table?

[✓ (a) 9/17] (b) 17/9 (c) 9 (d) 17 (e) none of these answers

$\frac{\#\text{elements}}{\text{space}}$

Use the following method to answer questions 2.4-2.7. This method performs a sequential search for a target recursively on an array of unique data values.

```
public int search(int[] data, int index, int target) {
     if (   stopping condition    ) return -1;
     else if (data[index] == target) return index;
     else return (    recursive call    );
}
```

DNE ✓

2.4 Assuming that this method is initially called with `index` = `0`, what is the correct stopping condition?

(a) `index == 0` (b) `index < 0`
[✓ (c)] `index == data.length` (d) `index > data.length`
(e) none of the answers above

2.5 Assuming that this method is initially called with `index` = `0`, what is the correct recursive call?

(a) `search(data, index-1, target)` backwards [✓ (b)] `search(data, index+1, target)`
(c) `search(data, 2*index+1, target)` (d) `search(data, index/2, target)`
(e) none of the answers above skipping elems skipping elems

2.6 If the array that we are searching has 64 values, what is the **minimum** number of recursive calls?
[✓ (a)] 0 (b) 1 (c) 63 (d) 64 ~~(e) none of these answers~~

10
target=8
→ |8|7|6|5|4|

2.7 If the array that we are searching has 64 values, what is the **maximum** number of recursive calls?
a) 6 (b) 63 [✓ (c)] 64 (d) 65 (e) none of these answers

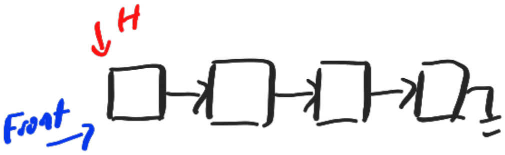
2.8 Consider the `IntArrayBag` class discussed in class. The following new `IntArrayBag` method supposedly determines if an instance of this class has the same number of integers as another instance of this class. What is wrong with this method?

```
public boolean sameSize(IntArrayBag otherBag)
{
     return (manyItems == otherBag.manyItems);
}
```

~~(a)~~ This method should be a `static` method since its parameter is of type `IntArrayBag`.
[✓ (b)] This method can throw a `NullPointerException` which is not indicated.
~~(c)~~ This method does not have direct access to the `manyItems` variable of the `otherBag` object.
~~(d)~~ This method should use the `equals` method to test for equality rather than the `==` operator.
(e) none of the answers above

2.9 Assuming that a queue is implemented using a singly linked list of `IntNode` nodes where `front` references the first node of the list only (there is no `rear` reference), what is the order of complexity of the `enqueue` operation if there are `n` nodes in the list?

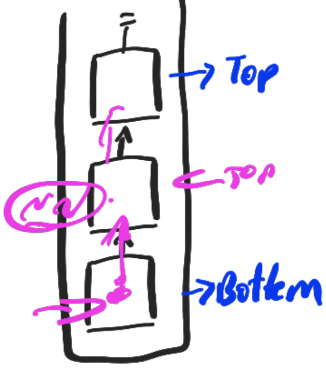
(a) O(1) [✓ (b)] O(n) (c) O(log n) (d) O(n^2) (e) none of these answers



USE THE FOLLOWING INFORMATION TO ANSWER PROBLEMS 2.10-2.11:

An `IntStack` is defined using a singly linked list of `IntNode` nodes such that the head of the list stores the bottom of the stack. The list has two variables, `bottom` and `top` which are references to the nodes with the bottom and top of the stack respectively.

```
  public void push(int value) {
1    IntNode newNode = new IntNode(value);
     if (top == null)
          bottom = newNode;
     else _a_;
     top = newNode;
  }
```



2.10 What is the correct expression for **a**?

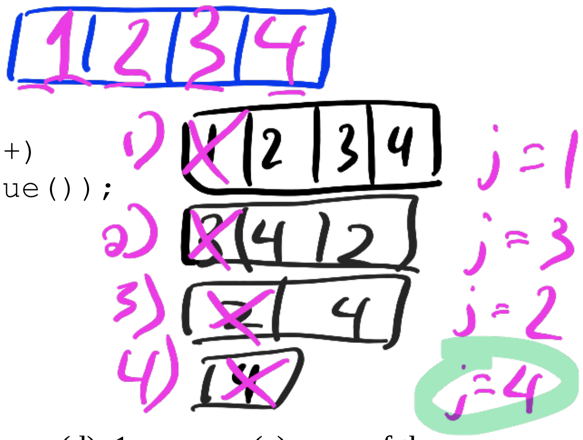
- (a) `top.setData(newNode);`
- [✓ (b) `top.setLink(newNode);`]
- (c) `newNode.setLink(top);`
- (d) `newNode.setData(top);`
- (e) none of these answers

2.11 If the operation `pop()` were implemented, what would be its worst case order of complexity if the stack was a list with n nodes?

(a) O(1) [✓ (b) O(n)] (c) O(n log n) (d) O(n^2) (e) none of these answers

USE THE FOLLOWING INFORMATION TO ANSWER PROBLEMS 2.12:

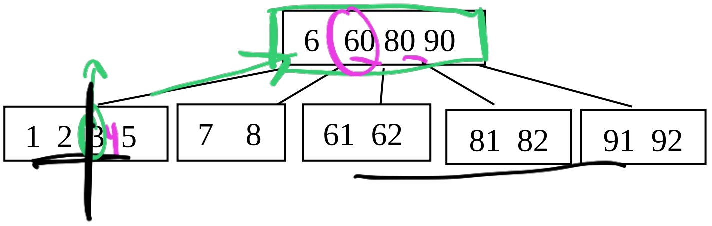
```
  public static int mystery(int n) {
     IntQueue q = new IntQueue();
     int i;
     int j = n;
     for (i = 1; i <= n; i++)
          q.enqueue(i);
     while (!(q.isEmpty())) {
          for (i = 1; i <= j; i++)
               q.enqueue(q.dequeue());
          j = q.dequeue();
     }
     return j;
  }
```

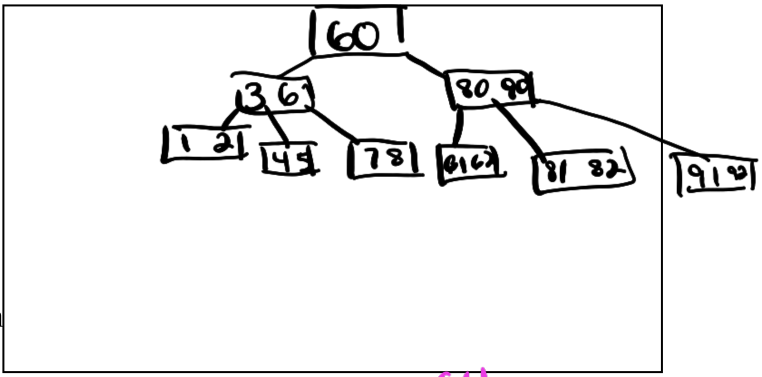


2.12 What does this method return if `n` = 4?

[✓ (a) 4] (b) 3 (c) 2 (d) 1 (e) none of these answers

4. Show the B-tree after the integer 4 is inserted into the following B-tree, where MINIMUM=2.





5. An array is sorted in an increasing order and contain

64 Elements

(a) If sequential search is used, what is the maximum number of comparisons that are needed to search for a target in this array? 64

(b) If binary search is used, what is the maximum number of comparisons that are needed to search for a target in this array? 7

elems. 64, 32, 16, 8, 4, 2, 1
1 2 3 4 5 6 7

(c) If the target is in position 0 of the array, which search technique would find the data faster? Why? Sequential

6. Trace how **selection** sort run on the following array of integers in an **increasing** order, showing the results after each run of the outer loop. Do not write a program.

10 21 8 18 14 5 70 1

1 21 8 18 14 5 70 10

1 5 8 18 14 21 70 10

1 5 8 10 14 21 70 18

1 5 8 10 14 18 70 21

1 5 8 10 14 18 21 70 Sorted! yay!

n [5 4 1 3 2]

↳ $\underset{O(n)}{\text{Array → heap}} + \underset{O(n \log n)}{\text{Remove from heap}} = \underset{O(n \log n)}{\text{Heap sort}}$