CS 225 Final Exam

Review Session

Contents

- MCQ Review

Linked Lists

Which of the following List ADT implementations gives us an O(1) time for `insertAtEnd`, i.e inserting an element at the end of the list?

I. A singly-linked list with only a `head` pointer.
II. A singly-linked list with `head` and `tail` pointers.
III. A doubly-linked list with only a `head` pointer.
IV. A doubly-linked list with `head` and `tail` pointers.

- (a) II and IV
- (b) None of the other options is correct
- (c) I, II, III and IV
- (d) I and III
- (e) I, III and IV

Linked Lists

Which of the following List ADT implementations gives us an O(1) time for `insertAtEnd`, i.e inserting an element at the end of the list?

I. A singly-linked list with only a `head` pointer.
II. A singly-linked list with `head` and `tail` pointers.
III. A doubly-linked list with only a `head` pointer.
IV. A doubly-linked list with `head` and `tail` pointers.

- (a) II and IV
- (b) None of the other options is correct
- (c) I, II, III and IV
- (d) I and III
- (e) I, III and IV

Tree Traversals

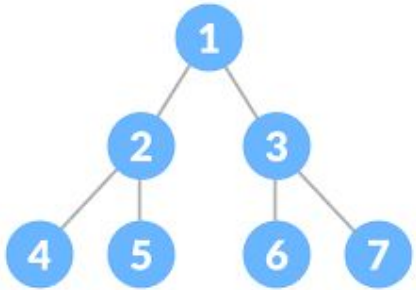
Among the following choices, which abstract data type should be used for a level order traversal of a binary tree?

- (a) hash table
- (b) array
- (c) linked list
- (d) queue
- (e) stack

Tree Traversals

Among the following choices, which abstract data type should be used for a level order traversal of a binary tree?

- (a) hash table
- (b) array
- (c) linked list
- (d) queue
- (e) stack



Tree Runtimes

Choose the appropriate running time from the list below.

The variable n represents the number of items (keys, data, or key/data pairs) in the structure. In answering this question you should assume the best possible implementation given the constraints, and also assume that every array is sufficiently large to handle all items (unless otherwise stated).

Perform an In-order traversal of a Binary Tree.

- (a) $O(1)$
- (b) $O(n \log n)$
- (c) $O(n)$
- (d) $O(n^2)$
- (e) $O(\log n)$

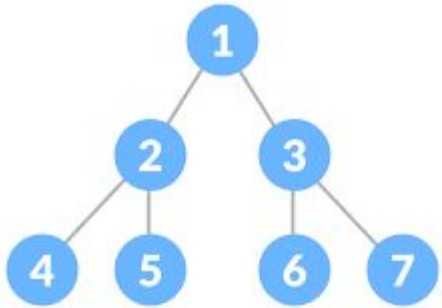
Tree Runtimes

Choose the appropriate running time from the list below.

The variable n represents the number of items (keys, data, or key/data pairs) in the structure. In answering this question you should assume the best possible implementation given the constraints, and also assume that every array is sufficiently large to handle all items (unless otherwise stated).

Perform an In-order traversal of a Binary Tree.

- (a) $O(1)$
- (b) $O(n \log n)$
- (c) $O(n)$
- (d) $O(n^2)$
- (e) $O(\log n)$



Disjoint Sets

Consider a disjoint set of **7** elements where every element initially starts as its own set. After performing the following operations **unioning by size with NO path compression**, what is the resulting array representation? In the event of a tie, you should assume the first input argument is 'larger in size'.

Union Steps:

```
union(0, 3)
union(4, 5)
union(6, 0)
union(5, 6)
```

Disjoint Sets

Consider a disjoint set of **7** elements where every element initially starts as its own set. After performing the following operations **unioning by size with NO path compression**, what is the resulting array representation? In the event of a tie, you should assume the first input argument is 'larger in size'.

Union Steps:

```
union(0, 3)
union(4, 5)
union(6, 0)
union(5, 6)
```

[-1, -1, -1, -1, -1, -1, -1] → [-2, -1, -1, 0, -1, -1, -1] → [-2, -1, -1, 0, -2, 4, -1]

→[-3, -1, -1, 0, -2, 4, 0] → [-5, -1, -1, 0, 0, 0, 0]

Heaps

For a minHeap implementation, assume we use the 0th index of the array to store the root (instead of index 1). Given an element at position i, what would be the position of its parent (assume $i \neq 0$)?

- (a) $\left\lfloor \frac{i}{2} \right\rfloor$
- (b) $\left\lfloor \frac{i-1}{2} \right\rfloor$
- (c) $\frac{i-1}{2}$
- (d) $\left\lceil \frac{i-1}{2} \right\rceil$
- (e) None of other options

Heaps

For a minHeap implementation, assume we use the 0th index of the array to store the root (instead of index 1). Given an element at position i, what would be the position of its parent (assume $i \neq 0$)?

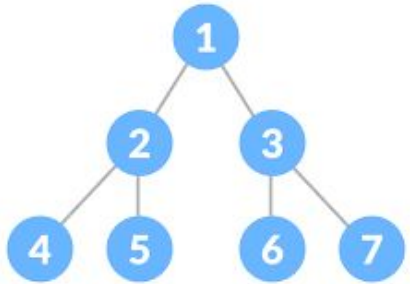
(a) $\left\lfloor \frac{i}{2} \right\rfloor$

(b) $\left\lfloor \frac{i-1}{2} \right\rfloor$

(c) $\frac{i-1}{2}$

(d) $\left\lceil \frac{i-1}{2} \right\rceil$

(e) None of other options



Bloom Filters

Consider the following bloom filter and hash functions. Assuming we used bloom filter `_find()`, which of the following queries would return 'true' (or 'present')?

To receive credit, you must correctly label all items as 'present' or 'absent'. (That is to say you must correctly check all 'true' items and NOT check any items which would return 'false')

Bloom Filter

```
[1,0,0,1,0,1,0,1,0,0]
```

Hashes

```
h1(x) = x % 10
h2(x) = (2*x + 3) % 10
```

- [ ] (a) `4`
- [ ] (b) `0`
- [ ] (c) `3`
- [ ] (d) `5`

Select all possible options that apply.

Bloom Filters

Consider the following bloom filter and hash functions. Assuming we used bloom filter `_find()`, which of the following queries would return 'true' (or 'present')?

To receive credit, you must correctly label all items as 'present' or 'absent'. (That is to say you must correctly check all 'true' items and NOT check any items which would return 'false')

Bloom Filter

```
[1,0,0,1,0,1,0,1,0,0]
```

Hashes

```
h1(x) = x % 10
h2(x) = (2*x + 3) % 10
```

- [ ] (a) 4
- [ ] (b) 0
- [ ] (c) 3
- [ ] (d) 5

Select all possible options that apply.