ads 辅学 lesson1

Zhejiang University, Advanced Data Structrue and Algorithm Analysis

赵一帆

2025 年 10 月 25 日

目录

avl tree and splay tre

判断题

For every AVL tree, there exists a sequence of nodes such that we can obtain this AVL tree by inserting the nodes in the sequence one by one into an initiallly empty tree.

avl tree and splay tre

判断题

For every AVL tree, there exists a sequence of nodes such that we can obtain this AVL tree by inserting the nodes in the sequence one by one into an initiallly empty tree.

引理：删去一棵 AVL 树的最深的叶子结点中的任意一个后，不需要进行旋转，这棵 AVL 树仍然是平衡的.

avl tree and splay tre

判断题

For a Splay tree T, $\Phi(T) = \sum_{i \in T} \log S_i$, where $S(i)$ is the number of descendants of i (i included). The insertion operation always causes the function $\Phi(T)$ to increase.

avl tree and splay tre

判断题

For a Splay tree T, $\Phi(T) = \sum_{i \in T} \log S_i$, where $S(i)$ is the number of descendants of i (i included). The insertion operation always causes the function $\Phi(T)$ to increase.

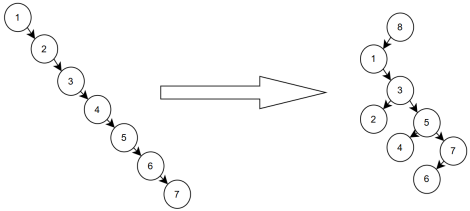
虽然结点增多，但是可能大多数节点的孩子数量减少.

avl tree and splay tre

判断题

For a Splay tree T, $\Phi(T) = \sum_{i \in T} \log S_i$, where $S(i)$ is the number of descendants of i (i included). The insertion operation always causes the function $\Phi(T)$ to increase.

虽然结点增多，但是可能大多数节点的孩子数量减少.



avl tree and splay tree

多选题

Among the following analyses of insertions and deletions of AVL trees, which is/are correct? We assume that "performing 1 rotation" means performing an LL, an LR, an RL or an RR rotation.

A.After inserting a node, we need to perform at most 1 rotation to rebalance the tree.

B.After deleting a node, we need to perform at most 1 rotation to rebalance the tree.

C.The time complexity of insertion is $O(1)$.

D.The time complexity of deletion is $O(1)$.

avl tree and splay tree

多选题

Among the following analyses of insertions and deletions of AVL trees, which is/are correct? We assume that "performing 1 rotation" means performing an LL, an LR, an RL or an RR rotation.
A.After inserting a node, we need to perform at most 1 rotation to rebalance the tree.
B.After deleting a node, we need to perform at most 1 rotation to rebalance the tree.
C.The time complexity of insertion is $O(1)$.
D.The time complexity of deletion is $O(1)$.

avl tree and splay tree

多选题

Consider the following statements on Splay trees. Select all the true statements.
A.splay(u) increases the rank of u by at most $\log(n)$.
B.During a splay operation, there can be at most one single rotation.
C.During a splay operation, there can be as many as $\Theta(n)$ rotations in the worst case.
D.The actual cost of each operation (findkey, insertion, deletion) of Splay trees is proportional to the number of rotations that this operation uses.

avl tree and splay tree

多选题

Consider the following statements on Splay trees. Select all the true statements.

A.splay(u) increases the rank of u by at most $\log(n)$.

B.During a splay operation, there can be at most one single rotation.

C.During a splay operation, there can be as many as $\Theta(n)$ rotations in the worst case.

D.The actual cost of each operation (findkey, insertion, deletion) of Splay trees is proportional to the number of rotations that this operation uses.

amortized cost

单选题

Consider the following buffer management problem. Initially the buffer size (the number of blocks) is one. Each block can accommodate exactly one item. As soon as a new item arrives, check if there is an available block. If yes, put the item into the block, induced a cost of one. Otherwise, the buffer size is doubled, and then the item is able to put into. Moreover, the old items have to be moved into the new buffer so it costs $k+1$ to make this insertion, where k is the number of old items. Clearly, if there are N items, the worst-case cost for one insertion can be $\Omega(N)$. To show that the average cost is $O(1)$, let us turn to the amortized analysis. Which of the following potential functions works?

A.The number of items currently in the buffer
B.The opposite number of items currently in the buffer
C.The number of available blocks currently in the buffer
D.The opposite number of available blocks in the buffer
E.The number of items currently in the buffer minus the number of available blocks in the buffer

amortized cost

单选题

Consider the following buffer management problem. Initially the buffer size (the number of blocks) is one. Each block can accommodate exactly one item. As soon as a new item arrives, check if there is an available block. If yes, put the item into the block, induced a cost of one. Otherwise, the buffer size is doubled, and then the item is able to put into. Moreover, the old items have to be moved into the new buffer so it costs $k+1$ to make this insertion, where k is the number of old items. Clearly, if there are N items, the worst-case cost for one insertion can be $\Omega(N)$. To show that the average cost is $O(1)$, let us turn to the amortized analysis. Which of the following potential functions works?

A.The number of items currently in the buffer

B.The opposite number of items currently in the buffer

C.The number of available blocks currently in the buffer

D.The opposite number of available blocks in the buffer

E.The number of items currently in the buffer minus the number of available blocks in the buffer

amortized cost

判断题 (2021 年期末考试)

Revisit the activity selection problem. Given a set of activities $S = a_1, a_2, \ldots, a_n$ that wish to use a resource, each a_i takes place during a time interval. The goal is to arrange as many compatible activities as possible. Recall that several greedy approaches are introduced in the class, among which the one selecting an activity with the shortest length, denoted by SF, is not always optimal. However, we claim that SF accepts at least $OPT/2$ activities, given that the optimal value is OPT, where OPT is an optimal solution. Check if the following is a correct proof.

We use a technique, called the charging scheme, similarly as the amortized analysis. Suppose each accepted activity of OPT holds one dollar, which will be given to the activities accepted by SF in the following way. For any activity a of OPT, if a is also accepted by SF, give the dollars to itself. Otherwise, there must be some activity a, accepted by SF, is not compatible with a. Then a receives one dollar from a. Along this line, each activity of OPT sends out one dollar to an activity in SF, while each activity of SF receives at most two dollars. It implies that SF accepts as least $OPT/2$ activities.

amortized cost

(1) $\widehat{c}_i = c_i + \Phi_i - \Phi_{i-1}$

(2) $\sum_i c_i = \sum_i \widehat{c}_i + \Phi_0 - \Phi_n$

amortized cost

(1) $\widehat{c}_i = c_i + \Phi_i - \Phi_{i-1}$

(2) $\sum_i c_i = \sum_i \widehat{c}_i + \Phi_0 - \Phi_n$

D 中刚 double 空间的时候有太多空格导致 ϕ_n 太小无法保证式子 (2)

amortized cost

(1) $\widehat{c}_i = c_i + \Phi_i - \Phi_{i-1}$

(2) $\sum_i c_i = \sum_i \widehat{c}_i + \Phi_0 - \Phi_n$

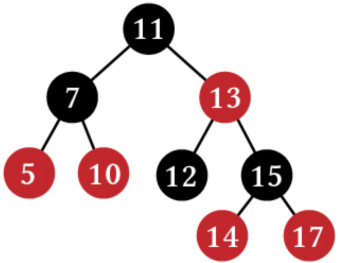
D 中刚 double 空间的时候有太多空格导致 ϕ_n 太小无法保证式子 (2)

E 中可以保证空格不会比装进去的 item 更多，所以 ϕ_n 始终非负

red-black tree and b-plus tree

判断题 (2022 年期末考试)

After deleting 12 from the following red-black tree, 13, 14, 15 and 17 will change its color



red-black tree and b-plus tree

判断题

A B+ tree of order 3 is also called a 2-3 tree. Consider a 2-3 tree with 3 internal nodes. Its leaves can have a maximum number of 18 keys in total.

red-black tree and b-plus tree

判断题

A B+ tree of order 3 is also called a 2-3 tree. Consider a 2-3 tree with 3 internal nodes. Its leaves can have a maximum number of 18 keys in total.

A m-order b-plus tree has at most m - 1 keys and m children at internal nodes ans at most m keys at leaf node.

red-black tree and b-plus tree

判断题

A B+ tree of order 3 is also called a 2-3 tree. Consider a 2-3 tree with 3 internal nodes. Its leaves can have a maximum number of 18 keys in total.

A m-order b-plus tree has at most m - 1 keys and m children at internal nodes ans at most m keys at leaf node.

Why?

red-black tree and b-plus tree

判断题

Consider an insertion in a B+ tree. We may need to update some keys stored in some internal nodes even if no leaf is split during the insertion.

red-black tree and b-plus tree

判断题

Consider an insertion in a B+ tree. We may need to update some keys stored in some internal nodes even if no leaf is split during the insertion.

新插入的值不可能被放到某个叶子结点的最左边一个键的位置, 因此内部结点不会有更新.

red-black tree and b-plus tree

判断题

Consider an initially empty B+ tree of order M. Whatever the value of M, after inserting n keys, the cost of a findkey operation on the resulting B+ tree is $\Theta(\log n)$.

red-black tree and b-plus tree

判断题

Consider an initially empty B+ tree of order M. Whatever the value of M, after inserting n keys, the cost of a findkey operation on the resulting B+ tree is $\Theta(\log n)$.

找到一个值一共需要访问 $\Theta(\log_M n) = \Theta(\log n)$ 个结点.（底数是常数，可以用换底公式换成任何常数底数.）因为 order M 是常数，因此在每个结点内无论采用线性搜索或二分搜索复杂度都是常数. 故而总的 findkey 时间复杂度为 $\Theta(\log n)$.

red-black tree and b-plus tree

单选题

Insert $1, 6, 7, 3, 5, 2$ one by one into an initially empty 2-3 tree (B+ tree of order 3). Which of the following statements is true? We assume that the height of a single node is 1.

A.The root has 1 key.

B.3 and 6 are in the same leaf.

C.The height of the resulting tree is 3.

D.The resulting tree is the same as that generated by inserting $1, 2, 3, 5, 6, 7$ one by one into an initially empty 2-3 tree.

red-black tree and b-plus tree

单选题

Insert $1, 6, 7, 3, 5, 2$ one by one into an initially empty 2-3 tree (B+ tree of order 3). Which of the following statements is true? We assume that the height of a single node is 1.

A.The root has 1 key.

B.3 and 6 are in the same leaf.

C.The height of the resulting tree is 3.

D.The resulting tree is the same as that generated by inserting $1, 2, 3, 5, 6, 7$ one by one into an initially empty 2-3 tree.

leftist heap and skew heap

判断题

A binary heap must be a Leftist heap.

leftist heap and skew heap

判断题

A binary heap must be a Leftist heap.

二叉堆是一种特殊的完全二叉树，它满足父节点的值不大于或不小于其子节点的值.

leftist heap and skew heap

判断题

Consider the concept of heavy and light nodes we introduced to analyze the Skew heaps. Now we "borrow" this concept to the Leftist heaps. We have that every node in a Leftist heap is light.

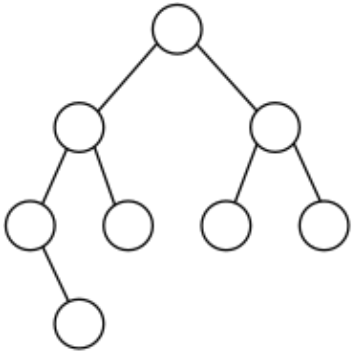
leftist heap and skew heap

判断题

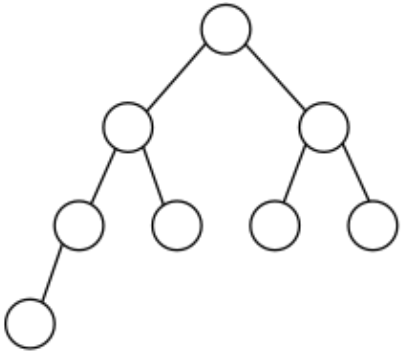
Consider the concept of heavy and light nodes we introduced to analyze the Skew heaps. Now we "borrow" this concept to the Leftist heaps. We have that every node in a Leftist heap is light.

- npl 和 heavy-light 的概念不同
- The null path length, Npl(X), of any node X is the length of the shortest path from X to a node without two children. Define Npl(NULL) = –1.
- A node p is heavy if the number of descendants of p' s right subtree is at least half of the number of descendants of p, and light otherwise. Note that the number of descendants of a node includes the node itself.

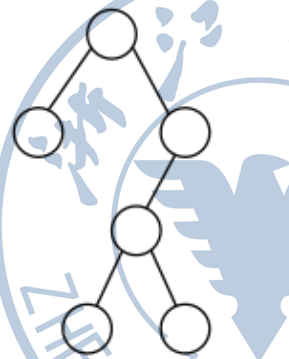
leftist heap and skew heap



(a)



(b)



(c)

leftist heap and skew heap

判断题

After inserting a node into a Leftist heap H (which is equivalent to merging a one-node Leftist heap with H), we need to swap the children of at most 1 node to make the resulting tree a Leftist heap.

leftist heap and skew heap

判断题

After inserting a node into a Leftist heap H (which is equivalent to merging a one-node Leftist heap with H), we need to swap the children of at most 1 node to make the resulting tree a Leftist heap.

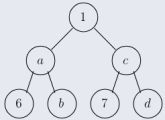
Assume that the node is inserted bewteen a_i and a_{i+1} in the right path.

- a_i is leaf, only rotate at a_i
- otherwise the npl of left child of a_i is at least 0
  - only need to rotate at the inserted node, because in this case, the inserted node is the right child of a_i, npl of which is 0.

leftist heap and skew heap

多选题

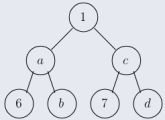
Consider the following **Leftist heap/skew heap** H. Assign $2, 3, 4, 5$ to a, b, c, d (in arbitrary order, as long as H keeps the heap property), and delete 1. Among the following options, which is/are the possible structure(s) of the resulting **Leftist heap/skew heap**?



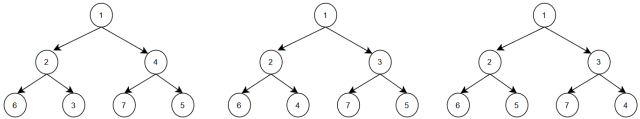
leftist heap and skew heap

多选题

Consider the following **Leftist heap/skew heap** H. Assign $2, 3, 4, 5$ to a, b, c, d (in arbitrary order, as long as H keeps the heap property), and delete 1. Among the following options, which is/are the possible structure(s) of the resulting **Leftist heap/skew heap**?



There are only 3 situations.



leftist heap and skew heap

选择题 (2023 年期末考试)

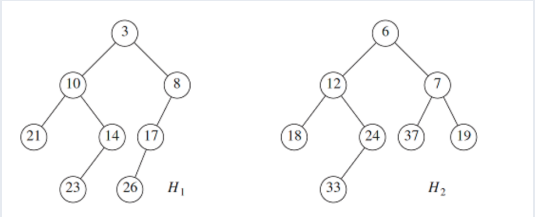
Merge the two skew heaps in the following figure. How many of the following statements is/are FALSE?

1.the null path length of 8 is the same as that of 12

2.18 is the left child of 12

3.21 is the right child of 10

A.0 B.1 C.2 D.3



leftist heap and skew heap

选择题（2023 年期末考试）

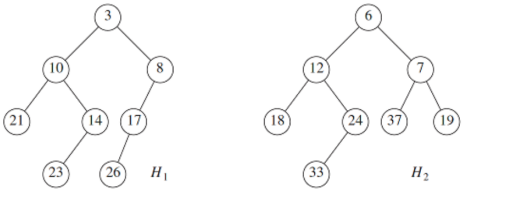
Merge the two skew heaps in the following figure. How many of the following statements is/are FALSE?

1.the null path length of 8 is the same as that of 12

2.18 is the left child of 12

3.21 is the right child of 10

A.0 **B.1** C.2 D.3



binomial queue

判断题

Consider a Binomial queue Q of size $n(n \geq 4)$, that is, there are n nodes in Q in total. We have that every Binomial tree in Q is a $\lfloor \log_2 n \rfloor$-ary tree.

binomial queue

判断题

Consider a Binomial queue Q of size $n(n \geq 4)$, that is, there are n nodes in Q in total. We have that every Binomial tree in Q is a $\lfloor \log_2 n \rfloor$-ary tree.

N 叉树是一种树形数据结构，其中每个节点最多可以有 N 个子节点.

binomial queue

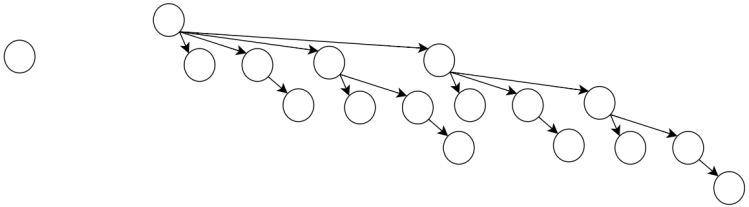
判断题

Consider a Binomial queue Q of size n, that is, there are n nodes in Q in total. Assume that there are k 1's in the binary representation of n. After deleting the minimum key of Q, we need to merge at most $k-1$ pairs of Binomial trees.

binomial queue

判断题

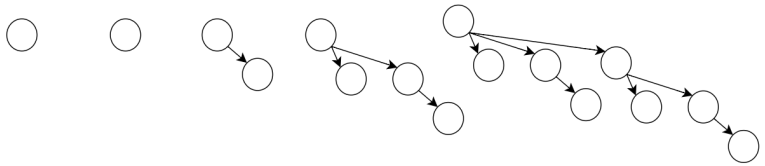
Consider a Binomial queue Q of size n, that is, there are n nodes in Q in total. Assume that there are k 1' s in the binary representation of n. After deleting the minimum key of Q, we need to merge at most $k-1$ pairs of Binomial trees.



binomial queue

判断题

Consider a Binomial queue Q of size n, that is, there are n nodes in Q in total. Assume that there are k 1' s in the binary representation of n. After deleting the minimum key of Q, we need to merge at most $k-1$ pairs of Binomial trees.



binomial queue

多选题

Right after we perform some operation (Merging, Insertion or DeleteMin) on a Binomial queue, we may have to merge some pairs of the resulting Binomial trees to make the resulting forest a Binomial queue. Suppose that $B_{i_1}, B_{i_2}, B_{i_3}$ of size $2^{k-1}, 2^{k-1}, 2^k (k \geq 1)$ respectively are Binomial trees to merge. Consider the case that B_{i_1} is merged with B_{i_2}, and the resulting Binomial tree is then merged with B_{i_3}. We call this case "cascading merge". Which of the following statements about "cascading merge" is/are correct?

A.We may have to perform "cascading merge" right after deleting the minimum key of a Binomial queue.

B.We may have to perform "cascading merge" right after merging two Binomial queues of the same size.

C.We must perform "cascading merge" right after inserting a key into a Binomial queue of odd size.

D.Consider the case that we perform consecutive insertions into a Binomial queue. Assume that we have performed "cascading merge" after inserting a key. Then in the next 3 insertions, we do not have to perform "cascading merge".

binomial queue

多选题

Right after we perform some operation (Merging, Insertion or DeleteMin) on a Binomial queue, we may have to merge some pairs of the resulting Binomial trees to make the resulting forest a Binomial queue. Suppose that $B_{i_1}, B_{i_2}, B_{i_3}$ of size $2^{k-1}, 2^{k-1}, 2^k (k \geq 1)$ respectively are Binomial trees to merge. Consider the case that B_{i_1} is merged with B_{i_2}, and the resulting Binomial tree is then merged with B_{i_3}. We call this case "cascading merge". Which of the following statements about "cascading merge" is/are correct?

A.We may have to perform "cascading merge" right after deleting the minimum key of a Binomial queue.

B.We may have to perform "cascading merge" right after merging two Binomial queues of the same size.

C.We must perform "cascading merge" right after inserting a key into a Binomial queue of odd size.

D.Consider the case that we perform consecutive insertions into a Binomial queue. Assume that we have performed "cascading merge" after inserting a key. Then in the next 3 insertions, we do not have to perform "cascading merge".

Backtracking

单选题

In the Tic-tac-toe game, a "goodness" function of a position is defined as $f(P) = W_X - W_O$, where W is the number of potential wins at position P. Player X tries to maximize the "goodness" function while player O tries to minimize it. Consider the following position. Now player O can choose one blank from a, b, c, d, e and f to play. which of the following statements is correct?

A.In this position, $W_X = 3$, $W_O = 2, f(P) = 1$.

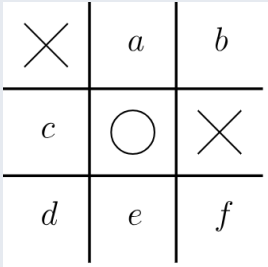
B,After player O plays a piece on b, d or e, the "goodness" function decreases by 2.

C.Player O has a strategy to win the game.

D.Suppose that player O plays a piece on a, c or f. For at least one of these three cases, player X has a strategy to win the game.

Backtracking

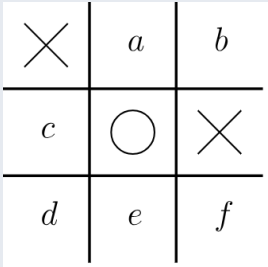
单选题



A.In this position, $W_X = 3$, $W_O = 2, f(P) = 1$.
B,After player O plays a piece on b, d or e, the "goodness" function decreases by 2.
C.Player O has a strategy to win the game.
D.Suppose that player O plays a piece on a, c or f. For at least one of these three cases, player X has a strategy to win the game.

Backtracking

单选题



A.In this position, $W_X = 3$, $W_O = 2, f(P) = 1$.
B,After player O plays a piece on b, d or e, the "goodness" function decreases by 2.
C.Player O has a strategy to win the game.
D.Suppose that player O plays a piece on a, c or f. For at least one of these three cases, player X has a strategy to win the game.

Backtracking

判断题

In the turnpike problem, if the longest distance between vertices and the total number of the vertices is given, the total sum of the series of distances is invariant.

多选题

In a turnpike reconstruction problem, the distance set is given as 1, 1, 2, 4, 4, 5, 5, 5, 6, 6, 7, 9, 10, 11, 12. In now backtracking state(a node in the backtracking tree), we temporarily identify four points: $x_1 = 0, x_2 = 12, x_3 = 1, x_4 = 2(x_4 = 10)$, which next try is possible ?
$A.x_5 = 3$ $B.x_5 = 4$ $C.x_5 = 5$ $D.x_5 = 6$ $E.x_5 = 7$ $F.x_5 = 8$ $G.x_5 = 9$