辅学 lesson3

Zhejiang University, Advanced Data Structrue and Algorithm Analysis

赵一帆

2025 年 11 月 29 日

目录

Mid-term exam

判断题 1

If sorting can be done in $O(n)$ time, then we can implement the Huffman's algorithm so that it runs in $O(n)$ time.

Mid-term exam

判断题 1

If sorting can be done in $O(n)$ time, then we can implement the Huffman's algorithm so that it runs in $O(n)$ time.

If we can execute huffman code algorithm on a sorted array in $O(n)$, then we can select true.

Mid-term exam

判断题 1

If sorting can be done in $O(n)$ time, then we can implement the Huffman's algorithm so that it runs in $O(n)$ time.

If we can execute huffman code algorithm on a sorted array in $O(n)$, then we can select true.
We can use "double queue" algorithm.

Mid-term exam

判断题 2

If the depth of an AVL tree with nodes $\{1, 2, 3, 4\}$ is 3(the depth of the root is 1), then it is possible for node 4 to be the root.

Mid-term exam

判断题 3

In an AVL tree, it is impossible to have this situation that the balance factors of a node and both of its children are all $+1$.

Mid-term exam

判断题 4

Insert {1, 2, 5, 3, 8, 4, -7, 10, 88, 34, 15, 63, 18, -18, 96} into an initially empty binomial queue, the resulting roots are 96, -18, -7 and 1.

Mid-term exam

判断题 4

Insert {1, 2, 5, 3, 8, 4, -7, 10, 88, 34, 15, 63, 18, -18, 96} into an initially empty binomial queue, the resulting roots are 96, -18, -7 and 1.

1, 2, 5, 3, 8, 4, -7, 10 is in a heap.
88, 34, 15, 63 is in a heap.
18, -18 is in a heap.
96 is in a heap.

Mid-term exam

判断题 5

The number of light nodes along the right path of a skew heap is $\Omega(logN)$.

Mid-term exam

判断题 5

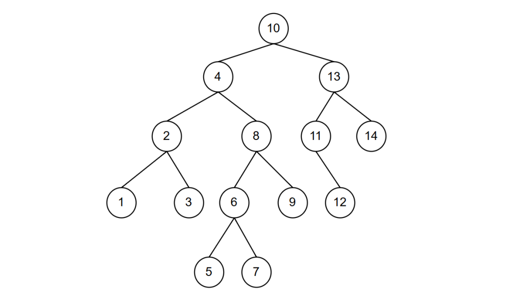
The number of light nodes along the right path of a skew heap is $\Omega(logN)$.

A degenerated skew heap have only heavy nodes except the leaf.

Mid-term exam

单选题 1

After deleting 14 from avl tree below, which of the following statements is incorrect?



A.The subtree rooted at 12 has 3 nodes B.4 and 10 are brothers
C.4 is father of 2 **D.The number of nodes which is in the subtree rooted at 4 increases**

Mid-term exam

单选题 2

Consider the following symbol frequencies for a five-symbol alphabet:

Symbol	Frequency
A	0.32
B	0.20
C	0.20
D	0.18
E	0.10

What is the average encoding length of an optimal prefix code?

A.2.10

B.2.28

C.2.48

D.3.00

Mid-term exam

单选题 3

Gvien the following table, what are the precision and the recall?

	Relevant	Irrelevant
Retrieved	3000	1000
Not Retrieved	21000	7000

What is the average encoding length of an optimal prefix code?

A.Precision = 75% and Recall = 12.5%

B.Precision = 12.5% and Recall = 75%

C.Precision = 75% and Recall = 87.5%

D.Precision = 87.5% and Recall = 75%

Mid-term exam

单选题 4

Suppose the running time $T(n)$ of an algorithm is bounded by the (non-standard!) recurrence with $T(1) = 1$ and $T(n) = T(\sqrt{n}) + 1$ for $n > 1$. Which of the following is the smallest correct upper bound on the asymptotic running time of the algorithm?

A. $O(1)$

B. $O(\log \log n)$

C. $O(\log n)$

D. $O(\sqrt{n})$

Mid-term exam

单选题 4

Suppose the running time $T(n)$ of an algorithm is bounded by the (non-standard!) recurrence with $T(1) = 1$ and $T(n) = T(\sqrt{n}) + 1$ for $n > 1$. Which of the following is the smallest correct upper bound on the asymptotic running time of the algorithm?

A. $O(1)$
B. $O(\log\log n)$
C. $O(\log n)$
D. $O(\sqrt{n})$

$$\frac{n}{k} \to \log n$$
$$\sqrt{n} \to \log\log n$$

Mid-term exam

单选题 4

Suppose the running time $T(n)$ of an algorithm is bounded by the (non-standard!) recurrence with $T(1) = 1$ and $T(n) = T(\sqrt{n}) + 1$ for $n > 1$. Which of the following is the smallest correct upper bound on the asymptotic running time of the algorithm?

A. $O(1)$

B. $O(\log\log n)$

C. $O(\log n)$

D. $O(\sqrt{n})$

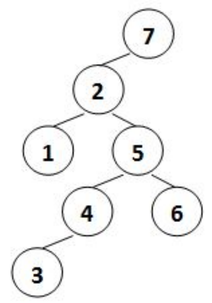
$\frac{n}{k} \to \log n$

$\sqrt{n} \to \log\log n$

The order of "right cost" is must higher than or equal to the number of $\Theta(1)$ in the last layer in the recursion tree.

Mid-term exam

单选题 5



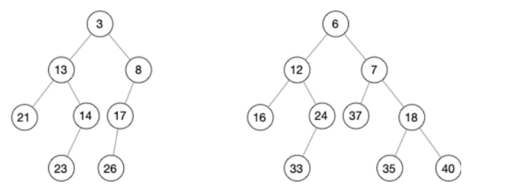
Which node is root 7's child after splaying 4 in the given splay tree?
A.2 B.4 **C.5** D.6

Mid-term exam

单选题 6

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

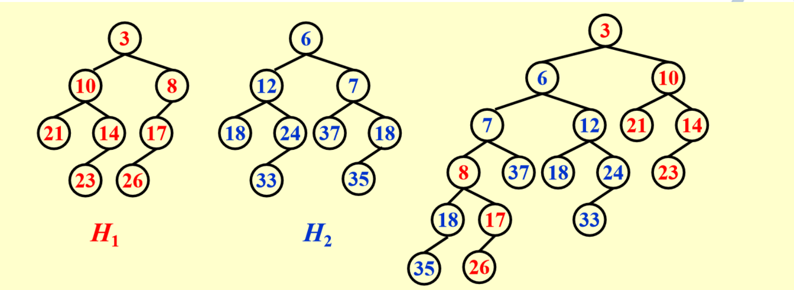
- the null path length of 8 is the same as that of 6
- 35 is the right child of 18
- the depths of 18 and 33 are the same



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

Mid-term exam

Always swap the left and right children except that the largest of all the nodes on the right paths does not have its children swapped.



Mid-term exam

多选题 1

Given an optimal prefix code for n symbols, which of the following statements are true? (Recall that the average code length is the sum of the code length of all symbols, weighted by their frequencies.)

A.The maximum coding length of a symbol can be $n-1$

B.The minimum code length of a symbol can be 1

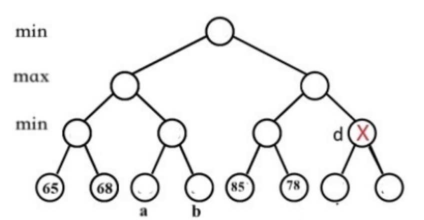
C.The average code length can be as large as $\Theta(n)$

D.The average code length can be as small as $O(1)$

Mid-term exam

多选题 2

Consider the following game tree. If node d is pruned by α-β pruning algorithm, which of the following statements about the value of node a or node b must be correct?



A.at least one of them are less than or equal to 78

B.at least one of them are greater than or equal to 78

C.both are less than or equal to 68

D.both are greater than or equal to 65

Mid-term exam

多选题 3

Start from N single-node splay trees, let's merge them into one splay tree in the following way: each time we select two splay trees, delete nodes one by one from the smaller tree and insert them into the larger tree. Then which of the following statements are true?

A.In any sequence of $N - 1$ merges, there are at most $O(N \log N)$ inserts.

B.Any node can be inserted at most $\log N$ times.

C.The amortized time bound for each insertion is $O(\log N)$.

D.The amortized time bound for each merge is $O(\log^2 N)$.

Mid-term exam

函数题

Suppose that a string of English letters is encoded into a string of numbers. To be more specific, A-Z are encoded into 0-25. Since it is not a prefix code, the decoded result may not be unique. For example, 1213407 can be decoded as BCBDEAH, MBDEAH, BCNEAH, BVDEAH or MNEAH. Note that 07 is not 7, hence cannot be decoded as H.Your job is to tell in how many different ways we can decode a numeric string.

int Decode(char NumStr[]);

Mid-term exam

函数题

Suppose that a string of English letters is encoded into a string of numbers. To be more specific, A-Z are encoded into 0-25. Since it is not a prefix code, the decoded result may not be unique. For example, 1213407 can be decoded as BCBDEAH, MBDEAH, BCNEAH, BVDEAH or MNEAH. Note that 07 is not 7, hence cannot be decoded as H.Your job is to tell in how many different ways we can decode a numeric string.

int Decode(char NumStr[]);

Dynamic programming: table entry and recursive function.

Backtracking

单选题

Assume that $T(1) = \Theta(1)$. Given $T(n) = 3T(\sqrt{n}) + \log n$, which of the following statements is correct?

F. $\log^{1.585} n$

Dynamic programming

单选题

In dynamic programming, we derive a recurrence relation for the solution to one subproblem in terms of solutions to other subproblems. To turn this relation into a bottom up dynamic programming algorithm, we need an order to fill in the solution cells in a table, such that all needed subproblems are solved before solving a subproblem. Among the following relations, which one is impossible to be computed?

A. $A(i,j)=\min(A(i-1,j),A(i,j-1),A(i-1,j-1))$

B. $A(i,j)=F(A(\min\{i,j\}-1,\min\{i,j\}-1),A(\max\{i,j\}-1,\max\{i,j\}-1))$

C. $A(i,j)=F(A(i,j-1),A(i-1,j-1),A(i-1,j+1))$

D. $\mathbf{A(i,j)=F(A(i-2,j-2),A(i+2,j+2))}$

Dynamic programming

多选题

Consider the knapsack problem of n items with capacity c. The weight and profit of item j are w_j and p_j respectively, $j = 1, 2, \ldots, n$. We have algorithms using dynamic programming to calculate the optimal solution. Here is a recursive formula of one of them.

$$z_j(d) = \begin{cases} z_{j-1}(d) & \text{if } d < w_j, \\ \max\{z_{j-1}(d), z_{j-1}(d - w_j) + p_j\} & \text{if } d \geq w_j. \end{cases}$$

In this formula, $z_j(d)$ denotes the optimal solution value considering only items $1, 2, \ldots, j$ with capacity d, where $j = 1, 2, \ldots, n$ and $d = 0, 1, \ldots, c$.

A. This algorithm is exact, that is, the result z^* of this algorithm is the optimal solution.

B. The worst-case running time of this algorithm is $\Theta(nc)$.

C. The worst-case running time of this algorithm is a polynomial in the length of the input.

D. The worst-case running time of this algorithm is a polynomial in the numeric value of the input, i.e. the largest integer present in the input.

Dynamic programming

多选题

A polygon（多边形）is convex（凸的）if and only if it is simple (i.e., it does not cross itself) and each interior angle is less than π. Consider a convex $n$gon（n 边形）. Choose $n-3$ diagonals（对角线）on this $n$gon that do not intersect except on the vertices（顶点）. These diagonals and the edges（边）of this $n$gon form $n-2$ triangles（三角形）. We call the set of these $n-3$ diagonals a triangulation（三角剖分）on this $n$gon. Let $f(n)$ denote the number of different triangulations of an n-gon, such as $f(4)=2$ because two different diagnoses divide the 4-gon into two different triangulations. Assume that $f(1)=f(2)=1$. Which are/is the correct dynamic programming method to calculate the $f(n)$?

A.$f(n)=f(\lfloor n/2\rfloor+1)+f(\lceil n/2\rceil+1)$.

B.$f(n)=\sum_{i=3}^{n-1}(f(i)+f(n-i+2))$.

C.$f(n)=\sum_{i=2}^{n-1}(f(i)\times f(n-i+1))$.

Dynamic programming

多选题

An optimal triangulation is one that minimizes the sum of some cost function of the $n-2$ triangles. Let $c(P)$ denote the cost of the optimal triangulation on a given n-gon $P = \langle v_0, v_1, \ldots, v_{n-1} \rangle$, where $v_0, v_1, \ldots, v_{n-1}$ are the vertices. We have that $c(\langle v_i, v_j, v_k \rangle)$ for $0 \le i < j < k \le n$ are constants and let $c(\langle v_{a_0}, v_{a_1}, v_{a_2}, \cdots v_{a_{m-1}} \rangle)$ denotes the optimal cost of m-gon $\langle v_{a_0}, v_{a_1}, v_{a_2}, \cdots v_{a_{m-1}} \rangle$. Let's take a convex quadrilateral（四边形） $P_4 = \langle v_0, v_1, v_2, v_3 \rangle$ as an example. We have that $c(P_4) = \max\{c(\langle v_0, v_1, v_2 \rangle) + c(\langle v_0, v_2, v_3 \rangle), c(\langle v_0, v_1, v_3 \rangle) + c(\langle v_1, v_2, v_3 \rangle)\}$. Which are/is the correct dynamic programming method to calculate the optimal cost in $O(n^3)$ time(n in the number of edge of the n-gon)?

Dynamic programming

多选题

A.For any sub-problem that we need to calculate the optimal cost of $c(\langle v_{a_0}, v_{a_1}, v_{a_2}, \cdots v_{a_{m-1}} \rangle)$, which $\{a_0, a_1, a_2, \cdots, a_{m-1}\} \subset \{1, 2, 3, \cdots, n\}$, we use the function $c(\langle v_{a_0}, v_{a_1}, v_{a_2}, \cdots, v_{a_{m-1}} \rangle) = \max_{i=0,\cdots,m-2} \{c(\langle v_{a_i \bmod m}, v_{a_{(i+1) \bmod m}}, v_{a_{(i+2) \bmod m}} \rangle) + c(\langle v_{a_i \bmod m}, v_{a_{(i+2) \bmod m}}, v_{a_{(i+3) \bmod m}}, \cdots, v_{a_{(i-1) \bmod m}} \rangle)\}$

B.We only need to calculate k-consecutive-point subproblem $c(\langle v_i, v_{i+1}, \cdots, v_j \rangle)$ (maybe $c(\langle v_i, v_{i+1}, \cdots, v_{n-1}, v_0, v_1, \cdots, v_j \rangle)$) using the function: $c(\langle v_i, v_{i+1}, \cdots, v_j \rangle) = \max_{k=i+1,i+2,\cdots,j-1} c(\langle v_i, v_j, v_k \rangle) + c(\langle v_i, v_{i+1}, v_{i+2}, \cdots, v_k \rangle) + c(\langle v_k, v_{k+1}, \cdots, v_j \rangle)$.

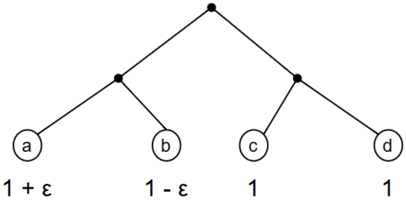
C.Given a base point a_0, we only need to calculate the subproblem $c(\langle v_0, v_i, v_{i+1}, \cdots, v_j \rangle)$ using the function

$$c(\langle v_0, v_i, v_{i+1}, \cdots, v_j \rangle) = \max_{k=i+1,i+2,\cdots,j-1} \{c(\langle v_0, v_i, v_{i+1}, \cdots, v_k \rangle) + c(\langle v_0, v_k, v_{k+1}, \cdots, v_j \rangle)\}.$$

Greedy

判断题

Consider an alphabet in which each symbol has a frequency. It is IMPOSSIBLE that there exists some optimal prefix encoding scheme that cannot be generated by the Huffman algorithm.



Greedy

单选题

Consider the problem of making change for n cents using the fewest number of coins. Assume that each coin's value is an integer. The coins of the lowest denomination（面额）is the cent.
(I) Suppose that the available coins are quarters (25 cents), dimes (10 cents), nickels (5 cents), and pennies (1 cent). The greedy algorithm always yields an optimal solution.
(II) Suppose that the available coins are in the denominations that are powers of c, that is, the denominations are $c_0, c_1, \ldots, c_k$, for some integers c>1 and k>=1. The greedy algorithm always yields an optimal solution.
(III) Given any set of k different coin denominations which includes a penny (1 cent) so that there is a solution for every value of n, greedy algorithm always yields an optimal solution.
Which of the following is correct?

C.Statement (III) is false.

Greedy

单选题

Consider the problem of making change for n cents using the fewest number of coins. Assume that each coin's value is an integer. The coins of the lowest denomination（面额）is the cent.
(I) Suppose that the available coins are quarters (25 cents), dimes (10 cents), nickels (5 cents), and pennies (1 cent). The greedy algorithm always yields an optimal solution.
(II) Suppose that the available coins are in the denominations that are powers of c, that is, the denominations are $c_0, c_1, \ldots, c_k$, for some integers c>1 and k>=1. The greedy algorithm always yields an optimal solution.
(III) Given any set of k different coin denominations which includes a penny (1 cent) so that there is a solution for every value of n, greedy algorithm always yields an optimal solution.
Which of the following is correct?
C.Statement (III) is false.

If the former always be the multiple of the latter, then greedy is optimal.

Greedy

单选题

Consider the problem of making change for n cents using the fewest number of coins. Assume that each coin's value is an integer. The coins of the lowest denomination（面额）is the cent.
(I) Suppose that the available coins are quarters (25 cents), dimes (10 cents), nickels (5 cents), and pennies (1 cent). The greedy algorithm always yields an optimal solution.
(II) Suppose that the available coins are in the denominations that are powers of c, that is, the denominations are $c^0, c^1, \ldots, c^k$, for some integers c>1 and k>=1. The greedy algorithm always yields an optimal solution.
(III) Given any set of k different coin denominations which includes a penny (1 cent) so that there is a solution for every value of n, greedy algorithm always yields an optimal solution.
Which of the following is correct?
C.Statement (III) is false.

If the former always be the multiple of the latter, then greedy is optimal.
For statement (I), we only need to consider 30, 35, 40, 45.

Problems

- In which situations, greedy is optimal in change-making?
- 3-dimension closest pair.
- OBST: $O(n^2)$?