R1-1 分数 2 作者 刘金飞 单位 浙江大学

There are 100000 documents in the database. The statistic data for one query is shown in the following table. The precision is 20%.

	Relevant	Irrelevant
Retrieved	10000	40000
Not Retrieved	30000	20000

◉ T ○ F

评测结果 答案正确

得分 2 分

R1-2 分数 2 作者 叶德仕 单位 浙江大学

Suppose we have a potential function Φ such that for all $\Phi(D_i) \geq \Phi(D_0)$ for all i, but $\Phi(D_0) \neq 0$. Then there exists a potential Φ' such that $\Phi'(D_0) = 0$, $\Phi'(D_i) \geq 0$ for all $i \geq 1$, and the amortized costs using Φ' are the same as the amortized costs using Φ.

◉ T ○ F

评测结果 答案正确

得分 2 分

R1-3 分数 2 作者 叶德仕 单位 浙江大学

Suppose that the replacement selection is applied to generate longer runs with a priority queue of size 3. Given the sequence of numbers { 81, 94, 11, 96, 12, 35, 17，99，28， 58，41，75，15}. Then 15 will belong to the 2nd run.

○ T ◉ F

评测结果 答案正确

得分 2 分

R1-4 分数 2 作者 陈昊 单位 浙江大学

In a Turnpike Reconstruction problem, given the distance set $D = \{1, 2, 2, 3, 3, 3, 5, 5, 6, 8\}$, there is a solution that does not include a point at position 6.

◉ T ○ F

评测结果 答案正确

得分 2 分

R1-5 分数 2 作者 陈昊 单位 浙江大学

To evaluate the prefix-sums of a sequence of 16 numbers by the parallel algorithm with Balanced Binary Trees, $C(1, 3)$ is found before $C(2, 1)$.

$C(h, i) = \sum_{k=1}^{a} A(k)$ where $(0, a)$ is the rightmost descendant leaf of node (h, i).

○ T ◉ F

评测结果 答案正确

得分 2 分

R1-6 分数 2 作者 丁尧相 单位 浙江大学

NP-complete problems are the hardest problems in NP-hard problems in terms of computational complexity.

○ T ◉ F

评测结果 答案正确

得分 2 分

R1-7 分数 2 作者 丁尧相 单位 浙江大学

For an approximation algorithm for a minimization problem, given that the algorithm does not guarantee to find the optimal solution, the best approximation ratio possible to achieve is a constant $\alpha > 1$.

○ T ◉ F

评测结果 答案正确

得分 2 分

R1-8 分数 2 作者 卜佳俊 单位 浙江大学

The height of an AVL tree of 30 nodes can be 5. (The height of an empty tree is defined to be -1)

◉ T ○ F

评测结果 答案正确

得分 2 分

R1-9 分数 2 作者 卜佳俊 单位 浙江大学

Considering leftist heaps of n nodes, the Merge operation has a higher time complexity than the DeleteMin operation.

○ T ◉ F

评测结果 答案正确

得分 2 分

R1-10 分数 2 作者 王灿 单位 浙江大学

Without any assumptions on the distances, if P $\neq$ NP, there is no ρ-approximation for TSP (Travelling Salesman Problem) for any $\rho \geq 1$.

○ T ◉ F

评测结果 答案错误

得分 0 分

R1-11 分数 2 作者 陈越 单位 浙江大学

Given a graph $G = (V, E)$. Let $A \subseteq V$ be any subset of V. If $(u, v) \in E$ is an minimum edge connecting A and $V - A$, then there exists an minimum spanning tree T of G such that $(u, v) \in T$.

○ T ◉ F

评测结果 答案错误

得分 0 分

R1-12 分数 2 作者 陈越 单位 浙江大学

A **Las Vegas** algorithm is a randomized algorithm that always gives the correct result, however the runtime of a Las Vegas algorithm differs depending on the input.
A **Monte Carlo** algorithm is a randomized algorithm whose output may be incorrect with a certain (typically small) probability. The running time for the algorithm is fixed however.
Then if a Monte Carlo algorithm runs in $O(n^2)$ time, with the probability 50% of producing a correct solution, then there must be a Las Vegas algorithm that can get a solution in $O(n^2)$ time in expectation.

◉ T ○ F

评测结果 答案错误

得分 0 分

R1-13 分数 2 作者 Yang Yang 单位 浙江大学

In the master method, we define $T(N)$ in the recursive form $T(N) = aT(N/b) + f(N)$. It means that the algorithm divides the problem into a parts, with each part being $\frac{1}{b}$ the size of the original. It costs $f(N)$ to gather results from subproblems and to derive its own result.

◉ T ○ F

评测结果 答案正确

得分 2 分

R2-1 分数 3 作者 刘金飞 单位 浙江大学

When searching for the keyword "game theory" on Google, the first four results returned are as follows:

Doc	Text
1	Game theory is the study of mathematical models of strategic interactions among rational agents. It has applications in many social science fields.
2	Game theory is the study of how players strategize and make decisions. It's a way to model scenarios in which conflicts of interest exist among the players.
3	Game theory is a branch of applied mathematics that provides tools for analyzing situations in which players make interdependent decisions.
4	Hello Internet! Welcome to GAME THEORY! If you are like us, then you have probably wondered about the secrets hidden in your favorite games.

Now if we construct an inverted file index for the results shown above, ignore case（忽略大小写）and do word stemming, please fill in the blanks in the inverted file index (not complete) below:

No.	Term	Times; Documents; Words
1	game	$\langle 5; (1;1), (2;1), (3;1), (4;5), (4;24) \rangle$
2	strategy	$\langle 2; (1;10), (2;①) \rangle$
3	apply	$\langle 2; (1;17), (3;7) \rangle$
4	math	$\langle 2; (1;7), (②;8) \rangle$
…	…	…

A. $① = 9, ② = 3$

B. $① = 9, ② = 1$

C. $① = 10, ② = 3$

D. $① = 10, ② = 1$

评测结果 答案正确

得分 3 分

R2-2 分数 3 作者 刘金飞 单位 浙江大学

If we insert $N (N \geqslant 2)$ nodes (with different integer elements) consecutively to build a red-black tree T from an empty tree, which of the following situations is possible:

A. All nodes in T are black

B. The number of leaf nodes (NIL) in T is $2N-1$

C. $2N$ rotations occurred during the construction of T

D. The height of T is $\lceil 3\log_2(N+1) \rceil$ (assume the height of the empty tree is 0)

评测结果 答案正确

得分 3 分

R2-3 分数 3 作者 叶德仕 单位 浙江大学

In external sorting, suppose we have n runs of lengths $2^0, 2^1, \ldots, 2^{n-1}$, respectively. To obtain the minimum merge time, which of the following statement is FALSE?

A. The first merge will be the run of lengths 2^0 and 2^1.

B. The total number of merges is $n-1$.

C. The order of merging must be ordered in the decreasing lengths.

D. The Huffman's algorithm will get the minimum merge time.

评测结果 答案正确

得分 3 分

R2-4 分数 3 作者 叶德仕 单位 浙江大学

A sum list L is a data structure that can support the following operations:

- Insert (x, L): insert the item x into the list L. The cost is 1 dollar.
- Sum(L): sum all items in the list L, and replace the list with a list containing one item that is the sum. The cost is the length of the list |L| dollars.

Now we would like to show that any sequence of Insert and Sum operations can be performed in $O(1)$ amortized cost per insert and $O(1)$ amortized cost per Sum. Which of the following statement is TRUE?

A. We use the accounting method that charges an amortized cost of 2 dollars to Insert and 0 dollar for Sum.

B. We use the potential function to be the number of elements in the list.

C. We use the potential function to be the opposite number of elements in the list.

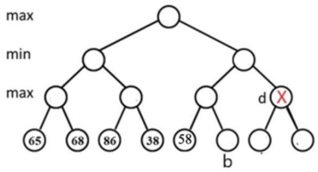
D. Neither method can show the amortized cost for Insert and Sum is $O(1)$.

评测结果 答案正确

得分 3 分

R2-5 分数 3 作者 陈昊 单位 浙江大学

Given the following game tree, node d will be pruned with $\alpha - \beta$ pruning algorithm if and only if ____.



- A. b $\le$ 68
- B. b $\ge$ 86
- C. b $\ge$ 58
- D. 58 $\le$ b $\le$ 68

评测结果 答案正确

得分 3 分

R2-6 分数 3 作者 陈昊 单位 浙江大学

The Merging problem is to merge two non-decreasing arrays A(1), A(2), ..., A(n) and B(1), B(2), ..., B(n) into another non-decreasing array C(1), C(2), ..., C($2n$). To solve it in parallel, we turn it into a Ranking problem. That is, to compute RANK(A(i),B) and RANK(B(i),A) for every $1 \le i \le n$, where RANK(e,S) is the position of `e` in `S`.

- The following psuedo-code is binary search parallel ranking for solving the problem parallely.

```
for Pi, 1<=i<=n  pardo
  RANK(A(i),B) := BS(A(i),B)
  RANK(B(i),A) := BS(B(i),A)
```

  where `BS(e,S)` is to find the position of `e` in `S` by binary search.

- The following psuedo-code is serial ranking for solving the problem.

```
i = j = 0;
while ( i ≤ n || j ≤ m ) {
  if ( A(i+1) < B(j+1) )
      RANK(++i, B) = j;
  else  RANK(++j, A) = i;
}
```

How many of the following statements are **True**?

- For binary search parallel ranking, $T(n) = O(\log n), W(n) = O(n \log n)$
- For serial ranking, $T(n) = O(n), W(n) = O(n)$
- Given a solution to the ranking problem, the merging problem can be solved in $O(1)$ time and $O(n)$ work.

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

评测结果 答案错误

得分 0 分

R2-7 分数 3 作者 陈昊 单位 浙江大学

For the Maximum Finding, how many of the following statements are **True**?

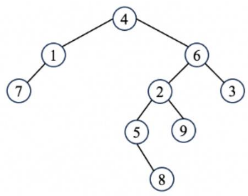
- Replace "+" by "max" in the parallel summation algorithm, we get an algorithm with $T(n) = O(\log n), W(n) = O(n)$.
- There exists a parallel algorithm solving the problem in constant time.
- With high probability, parallel random sampling algorithm can run with $O(\log n)$ work load.
- By adjusting the partition to $\sqrt{n}$ with a parallel divide and conquer paradigm, we can achieve $T(n) = O(\log \log n), W(n) = O(n)$

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

评测结果 答案正确

得分 3 分

R2-8 分数 3 作者 卜佳俊 单位 浙江大学



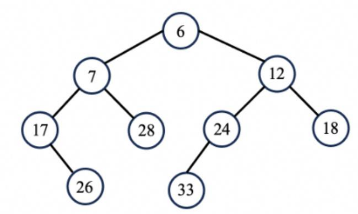
For the result of accessing the keys 2 and 3 (in order) of the splay tree in the figure, which one of the following statements is FALSE ? **(Note: the numbers are the indices of the nodes, not the key values.)**

- A. 3 is the root
- B. 1 is the parent of 7
- C. 4 and 9 are siblings
- D. 2 and 9 are siblings

评测结果 答案正确

得分 3 分

R2-9 分数 3 作者 卜佳俊 单位 浙江大学



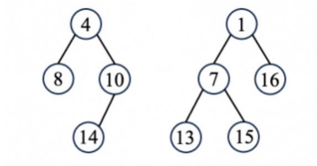
After inserting 15 into the Skew Heap in the figure, which of the following statements is correct for the resulted Skew Heap?

- A. Node 15 is the right child of node 12
- B. Node 17 is the right child of node 7
- C. Node 15 and node 24 are siblings
- D. The resulted heap is a leftist heap

评测结果 答案正确

得分 3 分

R2-10 分数 3 作者 卜佳俊 单位 浙江大学



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

- 8 and 10 are siblings
- along the left path from the root, we have 1, 4, 10, 16
- 15 is the right child of 7
- 4 and 10 have the same NPL

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

评测结果 答案正确

得分 3 分

R2-11 分数 3 作者 王灿 单位 浙江大学

Consider the metric **Facility Location** problem. A company wants to distribute its product to various cities. There is a set I of potential locations where a storage facility can be opened and a fixed "facility cost" f_i associated with opening a storage facility at location $i \in I$. (Sometimes we will say "facility i" to mean "facility at location i"). There also is a set J of cities to which the product needs to be sent, and a "routing cost" $c(i,j)$ associated with transporting the goods from a facility at location i to city j. The goal is to pick a subset S of I so that the total cost of opening the facilities and transporting the goods is minimized. In short, we want to minimize $C(S) = C_f(S) + C_r(S)$, where $C_f(S) = \sum_{i \in S} f_i$ and $C_r(S) = \sum_{j \in J} \min_{i \in S} c(i,j)$. Consider the following local serach algorithm:

1. Picking any subset S of the set I of facilities. This give us a feasible solution right away.
2. We then search for the local neighborhood of the current feasible solution to see if there is a solution with a smaller objective value; if we find one we update our feasible solution to that one.
3. We repeat step 2 until we do not find a local neighbor that yields a reduction in the objective value.

There are two types of "local steps" that we take in searching for a neighbor: i) remove/add a facility from/to the current feasible solution, or ii) swap a facility in our current solution with one that is not.

Which of the following statement is true?

- A. This algorithm is 2-approximation.
- B. Let S be a local optimum that the above algorithm terminates with, and S^* be a global optimum. Then $C_r(S) \geq C(S^*)$.
- C. Let S be a local optimum that the above algorithm terminates with, and S^* be a global optimum. Then $C_r(S) \leq C(S^*)$.
- D. None of the other options are correct.

评测结果 答案正确

得分 3 分

R2-12 分数 3 作者 丁尧相 单位 浙江大学

How many of the following arguments are correct?

- For non-deterministic Turing machine, "non-deterministic" means that the operations of the machine are random, such that the results of the operations are not deterministic.
- If a decision problem A can be reduced to B, then it means that problem A is strictly easier than B in terms of computational complexity.
- A decision problem in P is also in both NP and co-NP.

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

评测结果 答案正确

得分 3 分

R2-13 分数 3 作者 丁尧相 单位 浙江大学

Consider the 0-1 knapsack problem with object weights w, profits v, and total weight limit B (means that w of any object is no larger than B.). In the class, we have learned that combining the greedy algorithm on maximum profit v and maximum profit-weight ratio v/w leads to an approximation algorithm which always produces a solution no less than 1/2 of the optimal solution. Now let us consider the following simplified greedy algorithm. The algorithm first conducts the following sorting w.r.t. profit-weight ratio:

Sort all objects according to the profit-weight ratio $r_i = v_i/w_i$

so that $r_1 \geq r_2 \geq \cdots \geq r_n$.

Let the sorted order of objects be $a_1, \ldots, a_n$. The next step is to find the lowest k such that the total weight of the first k objects exceeds B. Finally, we pick the more valuable of $\{a_1, \ldots, a_{k-1}\}$ and $\{a_k\}$ as the final solution. Then which of the following arguments is correct:

- A. The algorithm always returns the optimal solution.
- B. The algorithm always returns a solution no less than α of the optimal solution, while $\alpha < 1/2$.
- C. The algorithm always returns a solution no less than $1/2$ of the optimal solution.
- D. The algorithm can generate a solution which is arbitrarily worse than the optimal solution.

评测结果 答案正确

得分 3 分

R2-14 分数 3 作者 陈越 单位 浙江大学

Which of the following is NOT an element of Greedy strategy?

- A. optimal substructure
- B. works only if the local optimum is equal to the global optimum
- C. overlapping sub-problems
- D. make a choice before solving the remaining sub-problem

评测结果 答案正确

得分 3 分

R2-15 分数 3 作者 陈越 单位 浙江大学

In a character set S consisting of 5 characters, given their occurrence frequencies being 3, 5, 7, 11 and 14, the weighted average length of Huffman codes for S is:

- A. 2.1
- B. 2.225
- C. 2.2
- D. 2.125

评测结果 答案正确

得分 3 分

R2-16 分数 3 作者 陈越 单位 浙江大学

To find the kth smallest number in a set S, randomized quick selection algorithm works in the following way:

1. If $|S| = 1$, then $k = 1$ and return the only element in S.
2. Randomly select a central splitter $p \in S$, which is a pivot that divides the set so that each side contains at least $|S|/4$ elements.
3. Partition $S - \{p\}$ into S_1 and S_2, as was done with quicksort.
4. If $k \leq |S_1|$, recursively find the kth smallest number in S_1. If $k = |S_1| + 1$, return the pivot as the answer. Otherwise, recursively find the $(k - |S_1| - 1)$st smallest number in S_2.

If $|S| = n$, then the best upper bound of the expected time complexity of this algorithm is:

- A. $O(n)$
- B. $O(n^2)$
- C. $O(n \log n)$
- D. $O((3/4)^n)$

评测结果 答案错误

得分 0 分

R2-17 分数 3 作者 Yang Yang 单位 浙江大学

Which of the following statements about binomial queue is FALSE?

- A. Find-Min operation can take $\Theta(1)$.
- B. The worst case of insertion is $\Theta(N)$.
- C. The amortized time of insertion is $\Theta(1)$.
- D. Delete-Min operation can take $\Theta(\log N)$.

评测结果 答案正确

得分 3 分

R2-18 分数 3 作者 Yang Yang 单位 浙江大学

We all know how to use the binary search method to find a value in a monotonic array. As an extension, we will use the ternary search method to find the extremum in a quadratic-function-like array. Formally, for an array a, let k be the position of its extremum. Then we have either $a[1] < ... < a[k-1] < a[k] > a[k+1] > ... > a[n]$ or $a[1] > ... > a[k-1] > a_k < a[k+1] < ... < a[n]$. Apply the ternary search method to locate k. For a given interval $1 \le l < r \le n$, locate the two trisection points in $[l, r]$ as $m1 = l + \lfloor \frac{r-l}{3} \rfloor$ and $m2 = l + \lfloor \frac{2(r-l)}{3} \rfloor$. Assuming we are looking for the minimum value, which of the following pieces of code can correctly narrow down the target interval?

- A. if(a[m1] < a[m2]) l = m1; else r = m2;
- B. if(a[m1] < a[m2]) l = m2; else r = m1;
- C. if(a[m1] < a[m2]) r = m2; else l = m1;
- D. if(a[m1] < a[m2]) r = m1; else l = m2;

评测结果 答案正确

得分 3 分

R2-19 分数 3 作者 Yang Yang 单位 浙江大学

In a binomial queue, we donote the total number of the nodes at even depth and odd depth as N_1 and N_2, respectively (the root is defined to be at the depth 0). Which of the following statements is FALSE?

- A. If $N_1 > N_2$, then $N_1 + N_2$ can be even.
- B. If $N_1 + N_2$ is odd, then $N_1 > N_2$.
- C. For all cases, $N_1 \ge N_2$.
- D. For all cases, $N_1 - N_2 \le 1$.

评测结果 答案正确

得分 3 分

R2-20 分数 3 作者 Yang Yang 单位 浙江大学

Consider the following function, where the time complexity for function `calc()` is $O(1)$.

```
void fun(int l, int r) {
    if(r-l+1<=1234) return;
    int m=(l+r)/2;
    int m1=(l+m)/2, m2=(m+1+r)/2;
    fun(l, m);
    fun(m1+1, m2);
    for(int k=1;k<=r-l+1;k++)
        for(int i=1;i<=r-l+1;i++)
            for(int j=l;j<=r;j+=i)
                calc(j, i);
    fun(m+1, r);
    fun(m1+1, m2);
}
```

Assume the initial input is `l=1, r=N`, What is the running time of this function? Your answer should be as tight as possible.

- A. $O(Nlog^2N)$
- B. $O(N^{2.5})$
- C. $O(N^2logN)$
- D. $O(N^2log^2N)$

评测结果 答案正确

得分 3 分

R5-1 分数 6 Is it a B+ tree? 作者 刘金飞 单位 浙江大学

The teacher wants to write the `IsBpT` function to check if the trees submitted by students satisfy the definition of the B+ tree of a given order (e.g., order 4) learned in our class. The B+ tree structure is defined as follows:

```
typedef struct BpTNode BpTNode;
struct BpTNode {
    bool isLeaf; /* 1 if this node is a leaf, or 0 if not */
    bool isRoot; /* 1 if this node is the root, or 0 if not */
    BpTNode** children; /* Pointers to children. This field is not used by leaf nodes. */
    ElementType* keys;
    int num_children; /* Number of valid children (not NULL) */
    int num_keys; /* Number of valid keys */
};
```

Fortunately, the students are all brilliant, so the B+ trees they submit guarantee to meet the following properties:

1. There is a root node, and all leaf nodes are at the same depth;
2. The key values stored in all leaf nodes are arranged in strictly ascending order from left to right.

Your task is to complete the function `IsBpT` as follows so that the teacher can determine whether a tree submitted by a student meets the other properties required by the definition of the B+ tree of a given order. Return `true` if the tree is a B+ tree, or `false` if not.

```
bool IsBpT(BpTNode* node, int order) {
    if (node->isLeaf == 1) { /* this is a leaf node */
        if (node->isRoot == 1) { /* this tree has only one node */
            if (node->num_keys < 1 || node->num_keys > order) return false;
        }
        else {
            if (node->num_keys < (order + 1) / 2 || node->num_keys > order) return false;
        }
    }
    else {
        /* check the property of the tree structure */
        if (node->num_keys != node->num_children - 1) return false;
        if (node->isRoot == 1) { /* this is the root node */
            if (node->num_keys < 1 || node->num_keys > order - 1) return false;
            else if (node->num_children < 2 || node->num_children > order) return false;
        }
        else {
            if ( node->num_keys < (order - 1) / 2  3分 || node->num_keys > order - 1) return false;
            else if (node->num_children < (order + 1) / 2 || node->num_children > order) return false;
        }

        /* check the property of the value of key */
        for (int i = 0; i < node->num_keys; i++) {
            BpTNode* key_node =  node->children[i+1]  3分 ;
            while (key_node->isLeaf == 0) {
                key_node = key_node->children[0];
            }
            if (node->keys[i] != key_node->keys[0]) return false;
        }
        for (int i = 0; i < node->num_children; i++) {
            if (IsBpT(node->children[i], order) == false) return false;
        }
    }
    return true;
}
```

评测结果 答案正确

得分 6 分

R6-1 Manager of Tasks 分数 8

全屏浏览 切换布局 作者 王灿 单位 浙江大学

There are N tasks arranged in a sequence on a machine waiting to be executed, and their order cannot be changed. You need to divide these N tasks into several groups, each containing several consecutive tasks. Starting from time 0, the tasks are processed in groups, and the time required to execute the i-th task is T_i. Additionally, the machine requires a startup time S before each group of tasks begins, so the time required for a group of tasks is the startup time S plus the sum of the time required for each task in this group.

After a task is executed, it will wait briefly in the machine until all tasks in that group are completely executed. That is to say, the tasks in the same group will be completed at the same time. The cost of each task is its completion time multiplied by a cost coefficient C_i.

Please plan a grouping scheme for the machine to minimize the total cost.

For all testing data, $1 \le N \le 1000, 0 \le S \le 50, 1 \le T_i, C_i \le 100$

Function Interface:

```
long long min_cost(int N, int S, int T[], int C[]);
```

where `T, C` are arrays of integers with `N` elements, and `S` is the startup time S mentioned above.

```
#include <stdio.h>
#define MAXN 1000
long long min_cost(int N, int S, int T[], int C[]);
int main() {
    int N, S;
    int T[MAXN], C[MAXN];
    scanf("%d%d", &N, &S);
    for (int i = 0;i < N; ++ i) {
        scanf("%d%d", &T[i], &C[i]);
    }
    printf("%lld\n", min_cost(N, S, T, C));
    return 0;
}
/* Your function will be put here */
```

Sample Input:

```
5
1
1 3
3 2
4 3
2 3
1 4
```

Sample Output:

```
153
```

Sample Explanation

We have grouped the tasks into 3 groups, which are `{1, 2}, {3}, {4, 5}`. The completion time corresponding to each task, in the order of the task numbers, is `{5, 5, 10, 14, 14}`. Similarly, the cost corresponding to each task, again in the order of the task numbers, is `{15, 10, 30, 42, 56}`. The total cost of these tasks is 153.

代码长度限制	16 KB
时间限制	400 ms
内存限制	64 MB