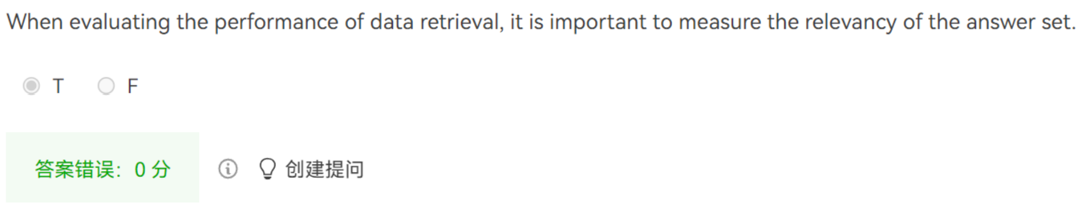
hw 错题整理

hw3

1.

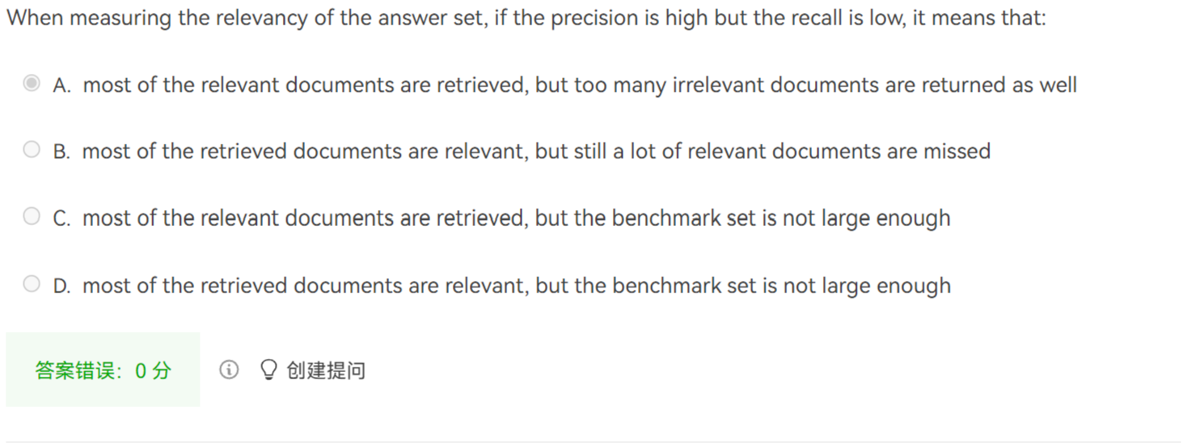


When evaluating the performance of data retrieval, it is important to measure the relevancy of the answer set.

● T ○ F

评价一个搜索引擎时需要measure relevancy,数据检索已经检索到了，就不需要评价了（文字游戏？？）

2.



When measuring the relevancy of the answer set, if the precision is high but the recall is low, it means that:

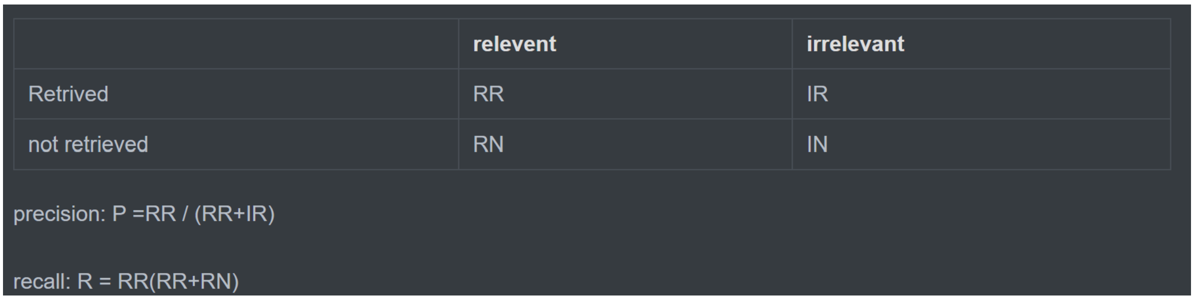
● A. most of the relevant documents are retrieved, but too many irrelevant documents are returned as well

○ B. most of the retrieved documents are relevant, but still a lot of relevant documents are missed

○ C. most of the relevant documents are retrieved, but the benchmark set is not large enough

○ D. most of the retrieved documents are relevant, but the benchmark set is not large enough

答案错误：0 分
创建提问



	relevent	irrelevant
Retrived	RR	IR
not retrieved	RN	IN

precision: P =RR / (RR+IR)

recall: R = RR(RR+RN)

precision(准确度)高而recall低，检测的文件中相关的多，但是很多相关文件还没有return

B

hw4

1. 6/9分

The function is to merge two leftist heaps H1 and H2.

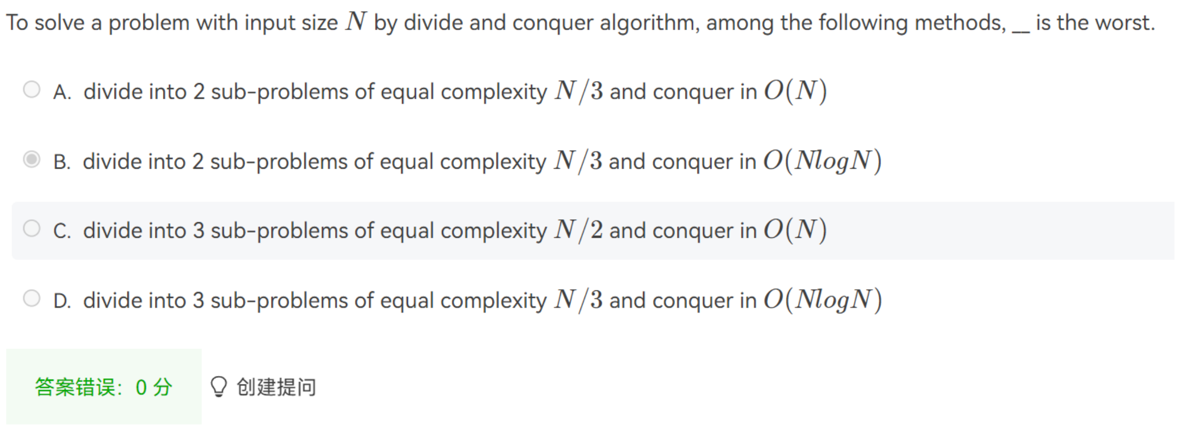
```
PriorityQueue Merge( PriorityQueue H1, PriorityQueue H2 ) {
if (H1==NULL) return H2;
if (H2==NULL) return H1;
    if (①)
swap(H1, H2);  //swap H1 and H2
if ( H1->Left == NULL )     ②
else {
H1->Right = Merge( H1->Right, H2 );
if ( H1->Left->Npl < H1->Right->Npl )
SwapChildren( H1 );  //swap the left child and right child of H1
③；  }
return H1; }
```

①：H1->Element>H2->Element

②：H1->Left = H2; //关键步骤！

③：H1->Npl=H1->Right->Npl+1; //更新Npl

hw7



To solve a problem with input size N by divide and conquer algorithm, among the following methods, __ is the worst.

○ A. divide into 2 sub-problems of equal complexity $N/3$ and conquer in $O(N)$

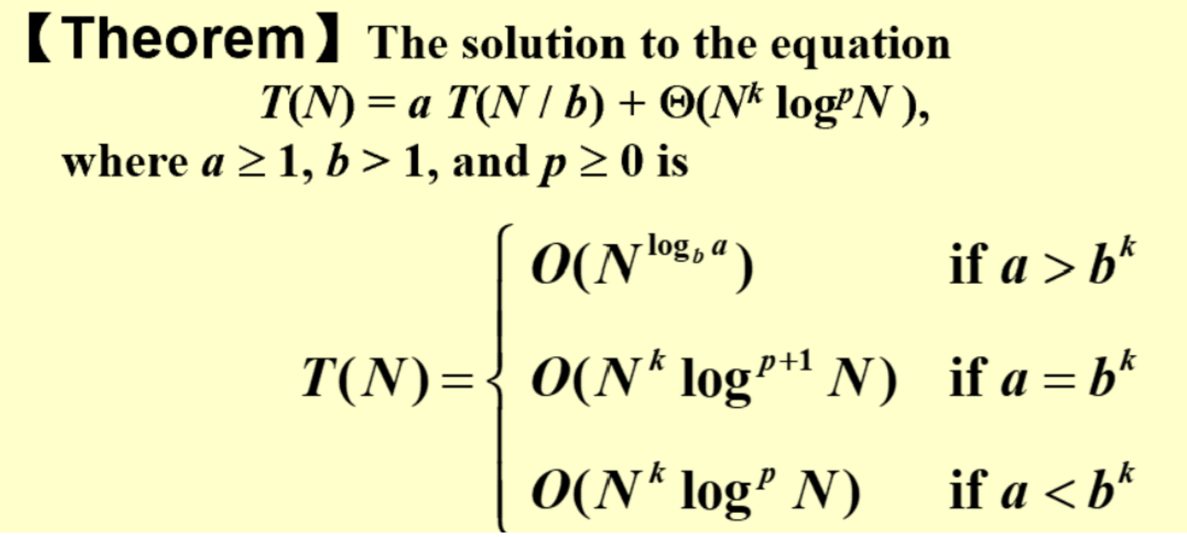
● B. divide into 2 sub-problems of equal complexity $N/3$ and conquer in $O(NlogN)$

○ C. divide into 3 sub-problems of equal complexity $N/2$ and conquer in $O(N)$

○ D. divide into 3 sub-problems of equal complexity $N/3$ and conquer in $O(NlogN)$

答案错误：0 分

解题方法：



【Theorem】The solution to the equation

$$T(N) = a\,T(N / b) + \Theta(N^k \log^p N),$$

where $a \ge 1$, $b > 1$, and $p \ge 0$ is

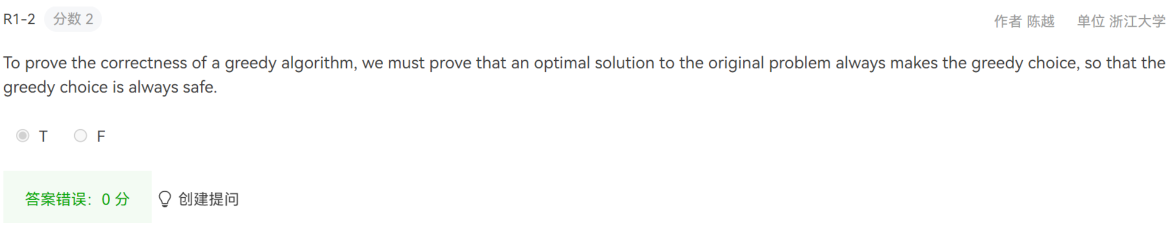
$$T(N) = \begin{cases} O(N^{\log_b a}) & \text{if } a > b^k \\ O(N^k \log^{p+1} N) & \text{if } a = b^k \\ O(N^k \log^p N) & \text{if } a < b^k \end{cases}$$

	a	b	k	p	$a \sim b^k$	result
A	2	3	1	0	$2 < 3^1$	$O(N)$
B	2	3	1	1	$2 < 3^1$	$O(N \log N)$
C	3	2	1	0	$3 > 2^1$	$O(N^{\log_2 3})$
D	3	3	1	1	$3 = 3^1$	$O(N \log^2 N)$

C. $O(N^{\log_2 3})$ 最大！

错误原因：误触还保存？你个憨憨

quiz9



R1-2 分数 2

作者 陈越 单位 浙江大学

To prove the correctness of a greedy algorithm, we must prove that an optimal solution to the original problem always makes the greedy choice, so that the greedy choice is always safe.

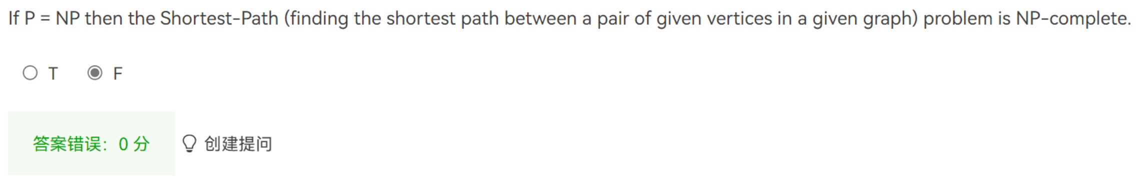
● T ○ F

答案错误：0 分

语言理解，证明贪心算法的正确性，需要证明局部最优是整体最优解的一部分，

而此题说最优解都是贪心的；应该改为贪心是最优的

quiz11 P-NP-NPC



If P = NP then the Shortest-Path (finding the shortest path between a pair of given vertices in a given graph) problem is NP-complete.

○ T ● F

答案错误：0 分

- P类问题：可以在多项式时间内求出问题的解的问题；
- NP问题： 可以在多项式时间内验证一个解是否正确的问题；如果一个问题是P类问题，它一定是NP问题；
- NPC问题：任何NP问题都可以规约(如果A问题可以在多项式时间内转化为B问题，就是说A可归约为B)

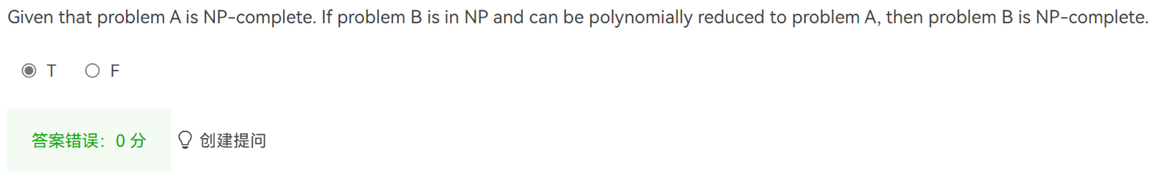
  为它的NP问题
- 千禧年难题:P=NP?

  解释：就是要证明或者证否 P和NP是否等价

  因为已知P类问题都是NP问题， P=NP? 转化为 证明或者证否 每个NP都是P;

  定理：如果找到了NPC问题的多项式时间解，则所有NP都有多项式时间解，且P=NP;

这道题目：如果P=NP，也就是说所有NP都有多项式时间解，所有的NP问题都是P,NP,NPC问题，全部等价

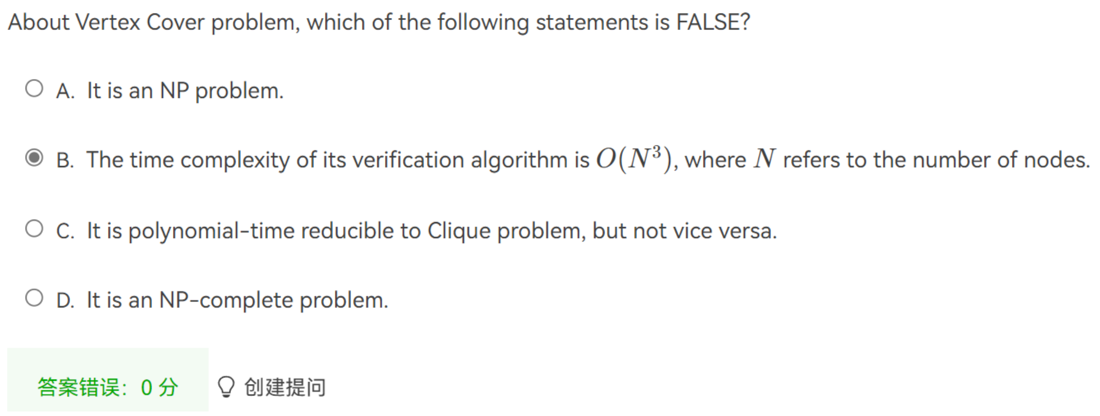


Given that problem A is NP-complete. If problem B is in NP and can be polynomially reduced to problem A, then problem B is NP-complete.

● T ○ F

答案错误：0 分

！！！！如果A是NPC问题，B是NP问题，A能规约到B，那么B是NPC问题！！！！顺序不能反!



About Vertex Cover problem, which of the following statements is FALSE?

○ A. It is an NP problem.

● B. The time complexity of its verification algorithm is $O(N^3)$, where N refers to the number of nodes.

○ C. It is polynomial-time reducible to Clique problem, but not vice versa.

○ D. It is an NP-complete problem.

答案错误：0 分

Vertex Cover 是NPC问题，和Clique问题相互归约，A,D正确，C错误；

B选项，遍历所有Node(N) 每一个node找到相邻节点（N) Vc集合中判断是否至少有其中一个(N)，那么时间复杂度就是 N^3

NPC问题及归约举例：

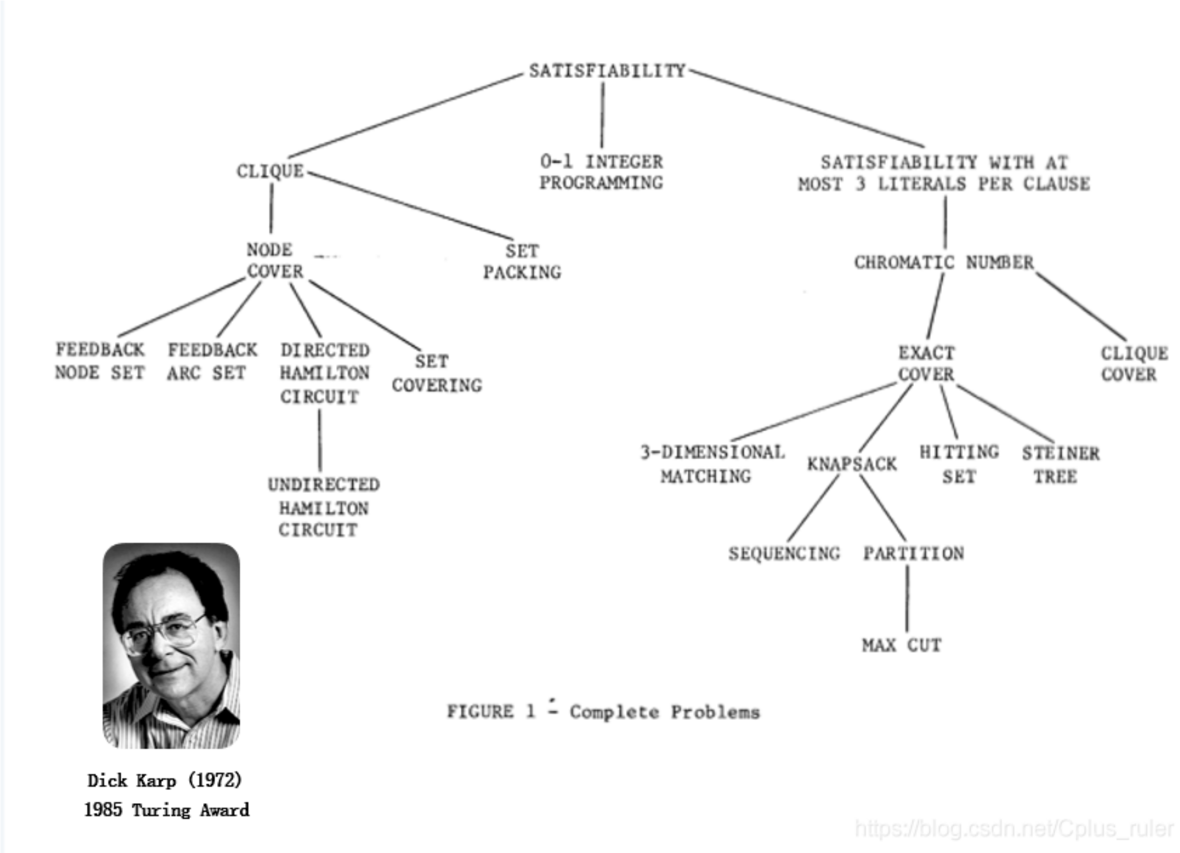
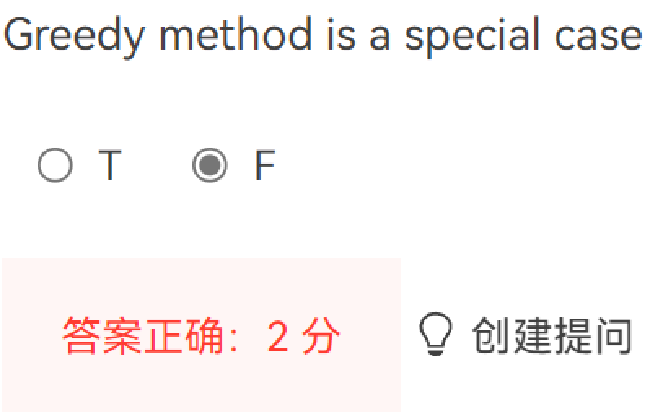


FIGURE 1 - Complete Problems

Dick Karp (1972)
1985 Turing Award

quiz 12



Greedy method is a special case of local search.

○ T ● F

答案正确：2 分

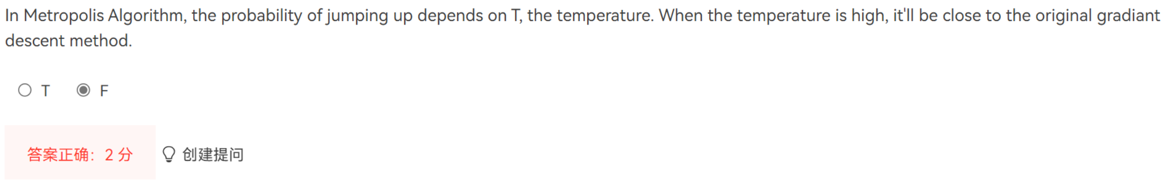
创建提问

贪心和local search 不一样

Me:
贪心算法与局部搜索的区别

贪心每一步根据启发信息的最优来决策，初始状态为零状态。而局部搜索则是从一个初始解中通过局部扰动，从而探索新解的可能。

贪心算法适用面比局部搜索更窄，贪心算法欲求得最优解的前提是满足最优子结构，而局部搜索即使满足最优子结构大概率之能够得到一个局部最优



In Metropolis Algorithm, the probability of jumping up depends on T, the temperature. When the temperature is high, it'll be close to the original gradiant descent method.

○ T ● F

答案正确：2 分

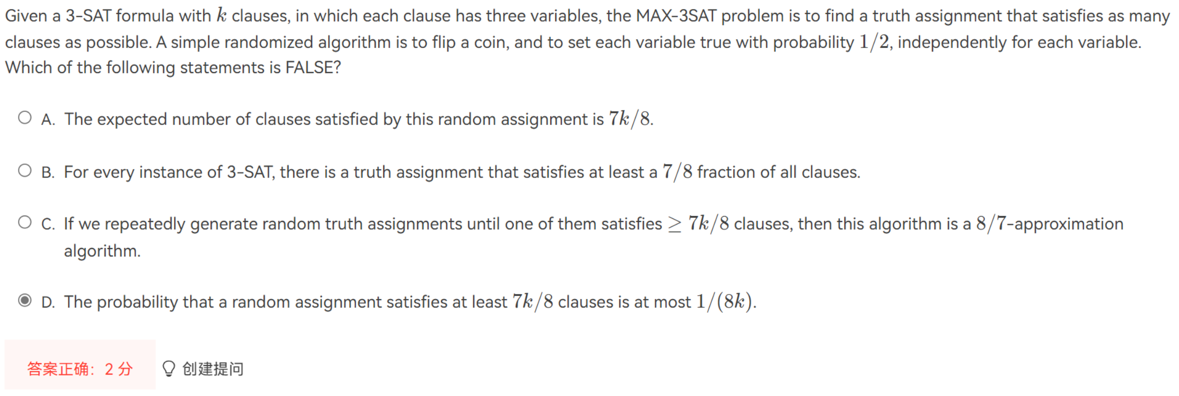
创建提问

temperature 高的时候可以绕过一些local optimization

hw12

太难了。。心力憔悴

quiz13



Given a 3-SAT formula with k clauses, in which each clause has three variables, the MAX-3SAT problem is to find a truth assignment that satisfies as many clauses as possible. A simple randomized algorithm is to flip a coin, and to set each variable true with probability $1/2$, independently for each variable. Which of the following statements is FALSE?

○ A. The expected number of clauses satisfied by this random assignment is $7k/8$.

○ B. For every instance of 3-SAT, there is a truth assignment that satisfies at least a $7/8$ fraction of all clauses.

○ C. If we repeatedly generate random truth assignments until one of them satisfies $\geq 7k/8$ clauses, then this algorithm is a $8/7$-approximation algorithm.

● D. The probability that a random assignment satisfies at least $7k/8$ clauses is at most $1/(8k)$.

答案正确：2 分　创建提问

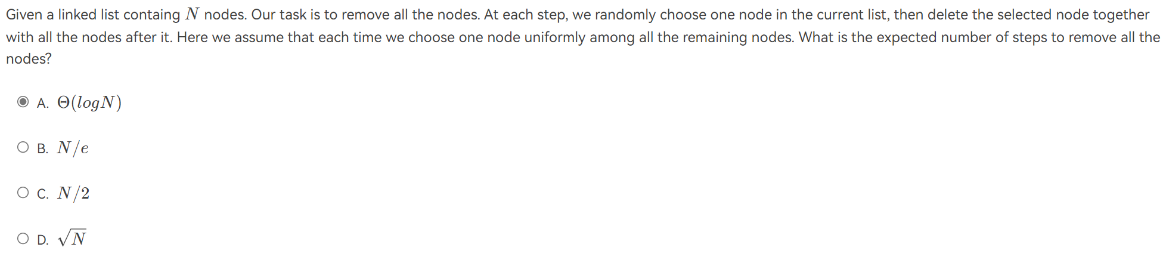
$(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_3 \vee x_4 \vee x_5)$

K- clause　1/8 false , 7/8 true.

$E(X) = \sum_{i=1}^{K} E(x_i) = 7/8 \cdot K.$

至少有一个方法使得 $E(x_i) > 7/8$, so 8/7 approximation.

hw13



Given a linked list containing N nodes. Our task is to remove all the nodes. At each step, we randomly choose one node in the current list, then delete the selected node together with all the nodes after it. Here we assume that each time we choose one node uniformly among all the remaining nodes. What is the expected number of steps to remove all the nodes?

● A. $\Theta(logN)$

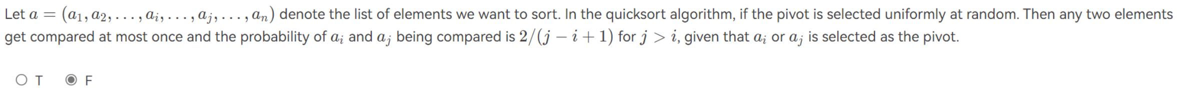
○ B. N/e

○ C. $N/2$

○ D. $\sqrt{N}$

布吉岛怎么算。。

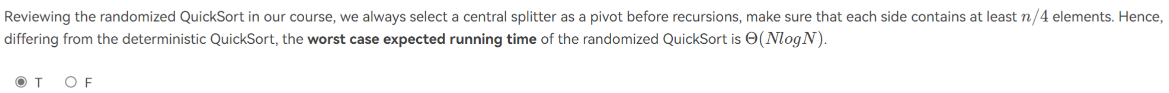
每一次从现有列表中去除的部分的期望是现有列表的一半，所以是logN



Let $a = (a_1, a_2, \ldots, a_i, \ldots, a_j, \ldots, a_n)$ denote the list of elements we want to sort. In the quicksort algorithm, if the pivot is selected uniformly at random. Then any two elements get compared at most once and the probability of a_i and a_j being compared is $2/(j-i+1)$ for $j > i$, given that a_i or a_j is selected as the pivot.

○ T　● F

这鬼题目是不是错了？pivot都比较一次？？



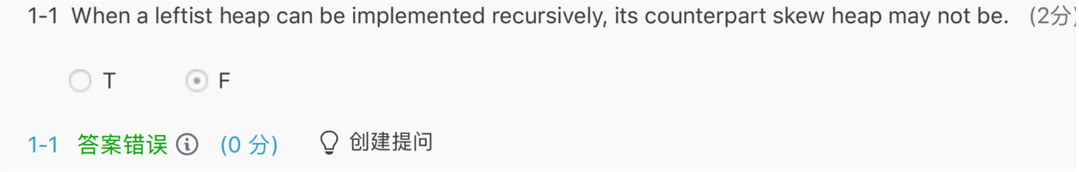
Reviewing the randomized QuickSort in our course, we always select a central splitter as a pivot before recursions, make sure that each side contains at least $n/4$ elements. Hence, differing from the deterministic QuickSort, the **worst case expected running time** of the randomized QuickSort is $\Theta(NlogN)$.

● T　○ F

期末

3. 记住答案就是了，skew heap确实对效率敏感的特殊输入，表现不大好，也不是用递归实现的

4.



1-1 When a leftist heap can be implemented recursively, its counterpart skew heap may not be. (2分

○ T ● F

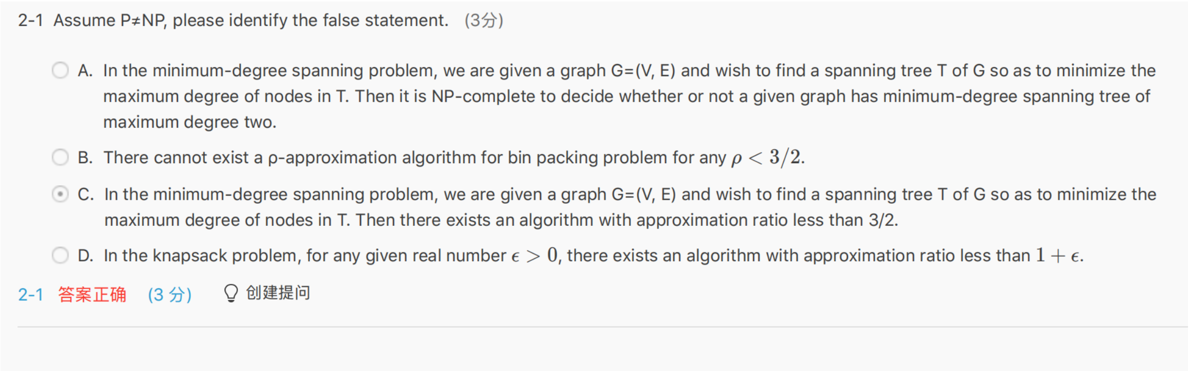
1-1 答案错误 ⓘ (0 分) 创建提问

4.

A.找到是否存在度最小的（度最大为2的）的spanning tree,肯定是NPC问题，参考哈密顿回路，旅行商问题啥的

C 你记住这个approximation ratio不可以低于3/2就对了

D 背包问题，用动态规划法是有最优解的，只是时间复杂度太大了(不是多项式时间)，要做一些约束



2-1 Assume P≠NP, please identify the false statement. (3分)

○ A. In the minimum-degree spanning problem, we are given a graph G=(V, E) and wish to find a spanning tree T of G so as to minimize the maximum degree of nodes in T. Then it is NP-complete to decide whether or not a given graph has minimum-degree spanning tree of maximum degree two.

○ B. There cannot exist a ρ-approximation algorithm for bin packing problem for any $\rho < 3/2$.

● C. In the minimum-degree spanning problem, we are given a graph G=(V, E) and wish to find a spanning tree T of G so as to minimize the maximum degree of nodes in T. Then there exists an algorithm with approximation ratio less than 3/2.

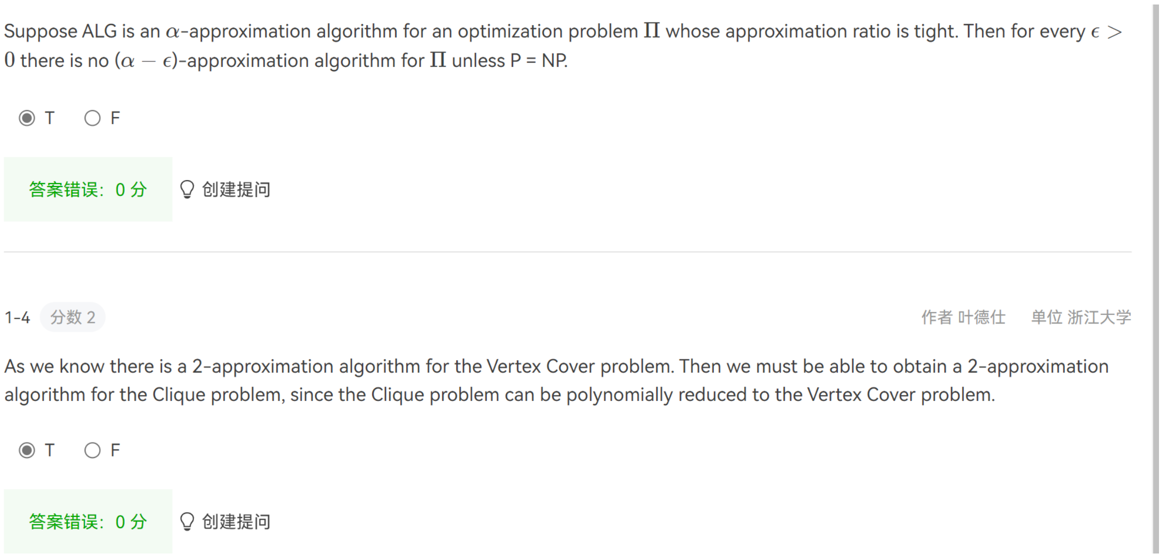
○ D. In the knapsack problem, for any given real number $\epsilon > 0$, there exists an algorithm with approximation ratio less than $1 + \epsilon$.

2-1 答案正确 (3 分) 创建提问

5. non-deterministic machine

All NP problems can be solved in polynomial time in a **non-deterministic machine**

6.记住答案先



Suppose ALG is an α-approximation algorithm for an optimization problem Π whose approximation ratio is tight. Then for every $\epsilon > 0$ there is no $(\alpha - \epsilon)$-approximation algorithm for Π unless P = NP.

● T ○ F

答案错误：0 分 创建提问

1-4 分数 2 作者 叶德仕 单位 浙江大学

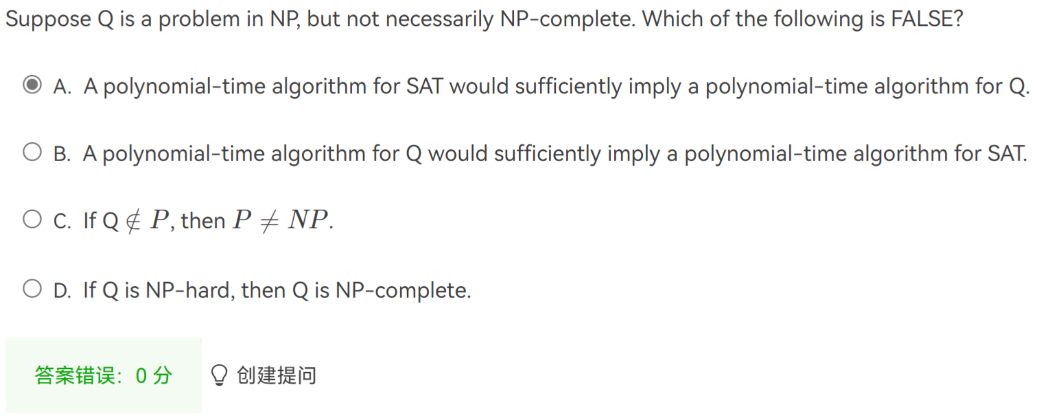
As we know there is a 2-approximation algorithm for the Vertex Cover problem. Then we must be able to obtain a 2-approximation algorithm for the Clique problem, since the Clique problem can be polynomially reduced to the Vertex Cover problem.

● T ○ F

答案错误：0 分 创建提问

7.B

A是对的，npc的问题的解可以解决np问题



Suppose Q is a problem in NP, but not necessarily NP-complete. Which of the following is FALSE?

● A. A polynomial-time algorithm for SAT would sufficiently imply a polynomial-time algorithm for Q.

○ B. A polynomial-time algorithm for Q would sufficiently imply a polynomial-time algorithm for SAT.

○ C. If Q $\notin P$, then $P \neq NP$.

○ D. If Q is NP-hard, then Q is NP-complete.

答案错误：0 分 创建提问

An NP-complete problem has the property that any problem in NP can be polynomially reduced to it.

不想看的题目增加了

2-5 ** Load balancing problem: ** We have n jobs $j = 1, 2, \ldots, n$ each with processing time p_j being an integer number. Our task is to find a schedule assigning n jobs to 100 identical machines so as to minimize the makespan (the maximum completion time over all the machines). (3分)

We adopt the following local search to solve the above load balancing problem.

**LocalSearch: *

Start with an arbitrary schedule.

Repeat the following until no job can be re-assigned:

- Let l be a job that finishes last.
- If there exists a machine i such that assigning job l to i allows l finish earlier, then re-assign l to be the last job on machine i.
- If such a machine is not unique, always select the one with the minimum completion time.

We claim the following four statements:

1. The algorithm LocalSearch finishes within polynomial time.
2. The Load-balancing problem is NP-hard.
3. Let OPT be the makespan of an optimal algorithm. Then the algorithm LocalSearch finds a schedule with the makespan at most of 1.95 OPT.
4. This algorithm finishes within $O(n^2)$.

How many statments are correct ?

○ A. 3

○ B. 4

◉ C. 2

○ D. 1

○ E. 0

2-5 答案错误 ⓘ (0 分) 创建提问