2-1 分数 3 作者 何钦铭 单位 浙江大学

Given 4 cases of frequences of four characters. In which case(s) that the total bits taken by Huffman codes are the same as that of the ordinary equal length codes?

- (1) 1 1 2 4
- (2) 1 2 3 4
- (3) 2 2 3 3
- (4) 2 2 3 4

- A. (3) only
- B. (3) and (4)
- C. (1), (2) and (4)
- D. None

2-2 分数 3 作者 叶德仕 单位 浙江大学

In Activity Selection Problem, we are given a set of activities $S = \{a_1, a_2, \ldots, a_n\}$ that wish to use a resource (e.g. a room). Each a_i takes place during a time interval $[s_i, f_i)$.

Let us consider the following problem: given the set of activities S, we must schedule them all using the minimum
number of rooms.

Greedy1:
Use the optimal algorithm for the Activity Selection Problem to find the max number of activities that can be scheduled in one room. Delete and repeat on the rest, until no activities left.

Greedy2:

- Sort activities by start time. Open room 1 for a_1.
- for $i = 2$ to n
  if a_i can fit in any open room, schedule it in that room;
  otherwise open a new room for a_i.

Which of the following statement is correct?

- A. None of the above two greedy algorithms are optimal.
- B. Greedy1 is an optimal algorithm and Greedy2 is not.
- C. Greedy2 is an optimal algorithm and Greedy1 is not.

○ D. Both of the above two greedy algorithms are optimal.

2-3 分数 2 作者 何钦铭 单位 浙江大学

There are n(=8) characters whose frequencies are the first n(=8) Fibonacci numbers. How many of the following statements about the Huffman Coding is/are correct?

(1) No more than 2 characters have the same code length.
(2) For any n $(n > 2)$, the maximum length of the codes is $n - 1$.
(3) The minimum length of codes is 2.

○ A. 0 ○ B. 1 ○ C. 2 ○ D. 3

2-4 分数 3 作者 叶德仕 单位 浙江大学

Which of the following statements is FALSE?

○ A. It is possible that NP-complete problem Vertex cover $\in P$ and Knapsack problem $\notin P$.

○ B. If P = NP then Shortest-Path is NP-complete.

○ C. For all problems $X \in NP$, $X \leq_p$ Bin-Packing, then Bin-Packing is NP-hard.

○ D. Any problem in P is polynomial-time reducible to any other problem in P.

2-5 分数 2 作者 张国川 单位 浙江大学

Assume that Problem X is reduced to Problem Y in polynomial time, where Y is NP-hard. Moreover, Y adimts a ρ-approximation algorithm, and there is no $(\rho - \epsilon)$-approroximation algorithm unless P=NP. Which one of the following statements is TRUE?

○ A. X is NP-hard too.

○ B. X has a ρ-approximation algorithm.

○ C. X has no $(\rho - \epsilon)$-approximation algorithm unless P=NP.

○ D. Neither of the other three is correct.

2-6 分数 2 作者 叶德仕 单位 浙江大学

Which of the following is **TRUE** about NP-Complete and NP-Hard problems?

○ A. If we want to prove that a problem X is NP-Hard, we take a known NP-Hard problem Y and reduce Y to X in polynomial time.

○ B. The first problem that was proved to be NP-complete was the circuit satisfiability problem.

○ C. NP-complete is a subset of NP-Hard

○ D. All of the other three.

2-7 分数 3 作者 叶德仕 单位 浙江大学

Which one of the following statements is FALSE?

○ A. A language L_1 is polynomial time transformable to L_2 if there exists a polynomial time function f such that $w \in L_1$ if $f(w) \in L_2$.

○ B. $L_1 \leq_p L_2$ and $L_2 \leq_p L_3$ then $L_1 \leq_p L_3$.

○ C. If $L_1 \in P$ then $L_1 \subseteq NP \cap$ co-NP.

○ D. If language L_1 has a polynomial reduction to language L_2, then the complement of L_1 has a polynomial reduction to the complement of L_2.

2-8 分数 4 作者 叶德仕 单位 浙江大学

Which of the following statement is true ?

○ A.Let $A$and B be optimization problems where it is known that A reduces to B in polynomial time. Additionally, suppose that there exists a polynomial-time 2-approximation for B. Then there must exist a polynomial time 2-approximation for A.

○ B. There exists a polynomial-time 2-approximation algorithm for the general Traveling Salesman Problem.

○ C.A randomized algorithm for a decision problem with one-sided-error and correctness probability 1/3 (that is, if the answer is YES, it will always output YES, while if the answer is NO, it will output NO with probability 1/3) can always be amplified（放大） to a correctness probability of 99%.

○ DSuppose that you have two deterministic online algorithms, A_1 and A_2, with competitive ratios (*the approximation ratio for an online algorithm is called competitive ratio*) c_1 and c_2 respectively. Consider the randomized algorithm A^* that flips a fair coin once at the beginning; if the coin comes up heads, it runs A_1 from then on; if the coin comes up tails, it runs A_2 from then on. Then the expected competitive ratio of A^* is at least $\min\{c_1, c_2\}$.

2-9　分数 4　　作者 叶德仕　　单位 浙江大学

K-center problem: Given N cities with specified distances, one wants to build K warehouses in different cities and minimize the maximum distance of a city to a warehouse.

Which of the following is false?

○ A. Given any constant $\alpha > 1$, unless P = NP, otherwise the K-center problem cannot be approximated within the factor α if the graph G admits an arbitrary distance function.

○ B. If the graph G obeys metric distance, then there is a 2-approximation algorithm for the K-center problem.

○ C. The K-center problem can be solved optimally in polynomial time if K is a given constant.

○ D. If the graph G obeys Euclidean distance, then there exists a PTAS for the K-center problem.

2-10　分数 3　　作者 Guochuan Zhang　　单位 浙江大学

Consider the bin packing problem which uses a minimum number of bins to accommodate a given list of items. Recall that Next Fit (NF) and First Fit (FF) are two simple approaches, whose (asymptotic) approximation ratios are 2 and 1.7, respectively. Now we focus on a special class $I2$ of instances in which only two distinct item sizes appear. Check which of the following statements is true by applying NF and FF on $I2$.

○ A. NF and FF both have improved approximation ratios.

○ B. NF has an improved approximation ratio, while FF does not.

○ C. FF has an improved approximation ratio, while NF does not.

○ D. Neither of NF or FF has an improved approximation ratio.

2-11　分数 3　　作者 沈鑫　　单位 浙江大学

How many of the following statements is/are **TRUE**?

- The 0-1 knapsack problem cannot be solved by any local search algorithm.
- The metropolis algorithm always improves the gradient descent algorithm.
- In some cases, the state-flipping algorithm cannot terminate.
- Unless $P = NP$, there is no ρ-approximation for the maximum cut problem for any $\rho < 2$.

○ A. 0

○ B. 1

○ C. 2

○ D. 3

2-12　分数 3　　　　作者 叶德仕　　单位 浙江大学

** 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).

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. 0

○ B. 1

○ C. 2

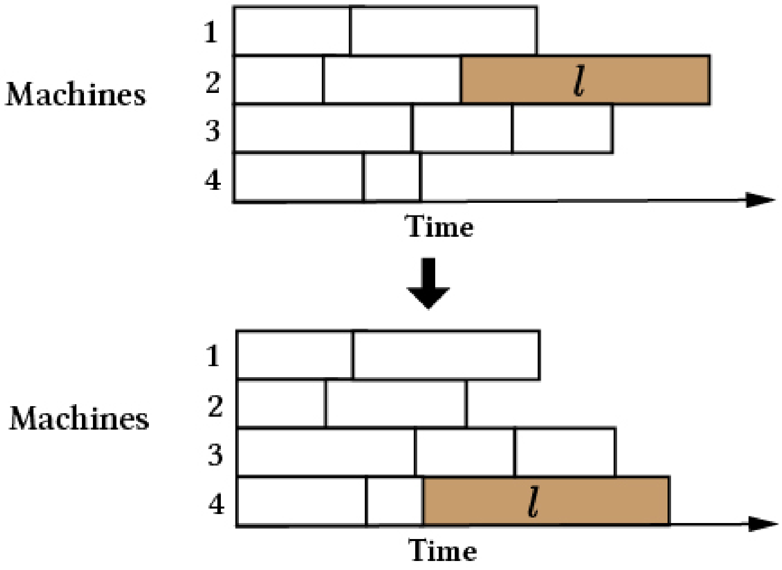
○ D. 3

○ E. 4

2-13　分数 1　　　　作者 yy　　单位 浙江大学

Scheduling Job Problem: There are n jobs and m identical machines (running in parallel) to which each job may be assigned. Each job $j = 1, \cdots, n$ must be processed on one of these machines for t_j time units without interruption. Each machine can process at most one job at a time. The aim is to complete all jobs as soon as possible; that is, if job j completes at a time C_j (the schedule starts at time 0), then we wish to minimize $C_{max} = max_{j=1,\cdots,n} C_j$. The length of an optimal schedule is denoted as $OPT(C_{max})$.

Local Search Algorithm: Start with any schedule; consider the job l that finishes last; check whether or not there exists a machine to which it can be reassigned that would cause this job to finish earlier. If so, transfer job l to this other machine. The local search algorithm repeats this procedure until the last job to complete cannot be transferred. An illustration of this local move is shown in following figure.



Which of the following statement is false?

- ○ A. $OPT(C_{max}) \geq \sum_{j=1}^{n} t_j/m$
- ○ B. When transfering a job, if we always reassign that job to the machine that is currently finishing earliest, then no job is transferred twice.
- ○ C. Upon the termination of the algorithm, the algorithm may return a schedule that has length at least $2OPT(C_{max})$
- ○ D.Suppose that we first order the jobs in a list arbitrarily, then consequently assign each job to the machine that is currently of earliest completion time, the schedule obtained cannont be improved by the local search procedure.