Assume standard complexity theoretic assumptions for all the questions.

Recall the PTAS for the problem of job scheduling over multiple parallel machines discussed in class. This involved a subroutine that would take a target T and any natural number k, and either output a schedule which obtains a makespan of $(1 + \frac{1}{k}) \cdot T$, or report that no schedule exists with makespan at most T. A job j is considered short if $p_j < \frac{T}{k}$. Which of the following is an INCORRECT step?

(a) Define a job j to be long if $p_j \geqslant \frac{T}{k}$, otherwise call j as short.

(b) The processing time of all jobs are rounded to nearest integral multiple of $\frac{T}{k}$.

(c) There are at most $(k+1)^{k^2}$ distinct configurations.

(d) The running time of the dynamic programming algorithm is $\mathcal{O}(n^{k^2} \cdot (k+1)^{k^2})$

b.

Which of the following best describes the **Bin Packing** problem?

(a) Given n items with sizes $s_1, s_2, \cdots, s_n$, and profits $p_1, p_2, \cdots, p_n$, for all $i \in [n]$, the computational task is to find a subset of items that maximizes the profit and fit in a bin of given size s.

(b) Given a set of n non-negative integers and a value s, the computational task is to print the subset of the given set whose sum is equal to the given value s.

(c) Given n items with sizes $s_1, s_2, \cdots, s_n$, and costs $c_1, c_2, \cdots, c_n$, for all $i \in [n]$, the computational task is to find a subset of items that minimizes the cost and also fits in an unit sized bin.

(d) Given n items with sizes $s_1, s_2, \cdots, s_n$, for all $0 \leqslant s_i \leqslant 1$, the computational task is to find a packing in unit-sized bins that minimizes the number of bins used.

d.

Which of the following is an INCORRECT statement?

(a) Knapsack admits no FPTAS.

(b) Bin Packing is an NP-hard problem.

(c) Scheduling jobs on multiple machines admits no FPTAS.

(d) There exists no $\rho > 0$, such that there exists a $(\frac{3}{2} - \rho)$ factor approximation algorithm for Bin packing problem.

a.

Consider the given instance to the **Bin Packing** problem in Table 1.

Item	Size
a_1	0.3
a_2	0.2
a_3	0.4
a_4	0.2
a_5	0.2
a_6	0.5
a_7	0.2

Table 1: Question 4

Which of the following is true about the instance?

(a) The minimum number of bins required is 3 but the instance is a no instance to the **Partition** problem.

(b) The minimum number of bins required is 2 but the instance is a no instance to the **Partition** problem.

(c) The minimum number of bins required is 2 and the instance is a yes instance to the **Partition** problem.

(d) The minimum number of bins required is 3 and the instance is a yes instance to the **Partition** problem.

c.

Recall the APTAS algorithm for **Bin Packing** discussed in the lecture. Which of the following is true?

(a) Any packing of large items of size greater than γ into l bins can be extended to a packing for the entire input with at least $\max\{l, \frac{1}{1-\gamma} \cdot \mathrm{SIZE}(I)+1\}$ bins, where $\mathrm{SIZE}(I) = \sum_{i=1}^{n} size_i$.

(b) If a new bin is opened, the minimum number of bins required to pack all items is at least $l+1$.

(c) If a new bin is opened, it can be observed that all of the existing bins have free space at least γ.

(d) An asymptotic polynomial-time approximation scheme (APTAS) is a family of algorithms $\{A_\varepsilon\}$ along with a constant c, where there exists an algorithm A_ε for each $\varepsilon > 0$ such that A_ε returns a solution of value at least $(1+\varepsilon)\mathrm{OPT} + c$ for minimization problems.

b.

What do you mean by a **preemptive** schedule?

(a) A schedule where jobs cannot be interrupted once they start execution.

(b) A schedule where jobs can be interrupted before their completion.

(c) A schedule where jobs are executed strictly based on their arrival time.

(d) A schedule where jobs are executed strictly based on their remaining processing time.

b.

Recall the **Shortest remaining processing time** algorithm with **non-preemptive** mode to schedule jobs on a single machine. Consider the releasing and processing time of the jobs given below. Compute the schedule and completion time for all jobs.

Job ID	Releasing time	Processing Time
J_1	1	7
J_2	2	5
J_3	3	1
J_4	4	2
J_5	5	8

Table 2: Question 7

(a) $< J_1, J_2, J_3, J_4, J_5 >$ and 23

(b) $< J_1, J_3, J_4, J_5, J_2 >$ and 24

(c) $< J_1, J_3, J_4, J_5, J_2 >$ and 23

(d) $< J_1, J_3, J_4, J_2, J_5 >$ and 24

d.

Consider the statements given below in context to scheduling jobs on single machine. Which of the following are true?

i. The goal is to minimize the average completion time for all jobs.

ii. A preemptive schedule can be computed in polynomial time that minimizes the completion time of all jobs.

iii. For a given set of n jobs, if a Shortest remaining processing time schedule completes a job j at C_j^P, then $\sum_{j=1}^{n} C_j^P$ is at least OPT.

iv. For a given set of n jobs, if a Shortest remaining processing time schedule completes a job j at C_j^P in preemptive mode and C_j^N in non-preemptive mode then $\sum_{j=1}^{n} C_j^P \leqslant 2 \cdot \sum_{j=1}^{n} C_j^N \leqslant 2 \cdot \text{OPT}$.

v. There exists a 2-factor approximation algorithm to compute the non-preemptive schedule.

(a) Only i, ii, iv and v are true

(b) Only ii, iii, and iv are true

(c) Only i, ii, and v are true

(d) All i, ii, iii, iv and v are true

c.

Recall the 3-approximation algorithm for scheduling jobs on a single machine with release dates to minimize the sum of weighted completion times discussed in the lecture. Which of the following is correct?

(a) $C_j^* \geqslant \frac{1}{2} p([j])$

(b) $\max_{k=1,\ldots,j} r_k + \sum_{k=1}^{j} p_k \leqslant C_j^N$

(c) $C_j^N + p([j]) \leqslant \max_{k=1,\ldots,j} r_k \leqslant 3 \cdot C_j^*$

(d) $C_j^* \sum_{k \in S} p_k < C_j^* \cdot p(S) = \sum_{k \in S} p_k C_k^*$

a.

Consider the Integer Linear Programming given below:

$$\text{minimize} \quad \sum_{i=1}^{m} y_i$$

subject to:

$$\sum_{i=1}^{m} x_{ij} = 1, \quad \forall j \in \{1, 2, \dots, n\}$$

$$\sum_{j=1}^{n} w_j x_{ij} \leqslant y_i, \quad \forall i \in \{1, 2, \dots, m\}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in \{1, 2, \dots, m\}, \forall j \in \{1, 2, \dots, n\}$$

$$y_i \in \{0, 1\}, \quad \forall i \in \{1, 2, \dots, m\}$$

The above integer linear programming is a formulation of

(a) 0-1 Knapsack problem

(b) Scheduling jobs on single machine

(c) Minimizing sum of weighted completion times

(d) Bin packing

d.