Homework 10

Thomas Kim tsk389 51835

David Munoz dam2989 51840

CS331 Algorithms and Complexity

Problem Q1(a). Given a collection of intervals on a time-line, and a bound k, does the collection contain a subset of nonoverlapping intervals of size at least k?
Claim: Interval Scheduling $\leq_p$ Vertex Cover
True

Proof. Vertex Cover is NP-Complete
Interval Scheduling is NP-Complete by the problem statement.
Therefore, Interval Scheduling is in NP.
All problems in NP are reducible to any NP-Complete problem.
Therefore, Interval Scheduling $\leq_p$ Vertex Cover.

□

Problem Q1(b). Claim: Independent Set $\leq_p$ Interval Scheduling

Proof. Independent Set is NP-Complete.
Therefore, Independent Set is in NP
By the problem statement, Interval Scheduling is NP-Complete
Therefore, all problems in NP can be reduced to Interval Scheduling in polynomial time.
Therefore, Independent Set $\leq_p$ Interval Scheduling

□

Problem Q2. Is the set of rational numbers countably infinite or uncountably infinite?
Countably Infinite

Proof. Let $r \in \mathbb{Q}$ be a rational number.
Consider rational numbers of the form $\frac{a}{b}$ where $a \in \mathbb{Z} \wedge b \in \mathbb{Z}_{\geq 0} \wedge gcd(a,b) = 1$
Claim: All rational numbers $\frac{q}{r}, q, r \in \mathbb{Z}$ can be expressed in this form.
First, factor out -1 from all negative q, r, such that the number is expressed $s\frac{|q|}{|r|}$ where $s = (-1)^1 \vee s = (-1)^2$
Next, divide q, r by $gcd(q,r)$ such that the number is expressed $s\frac{|q'|}{|r'|}$ where q', r' are the result after dividing q, r by $gcd(q,r)$
By the definition of gcd, q', r' share no common prime factors
Now, define a mapping $f : \mathbb{Q} \to \mathbb{Z} \times \mathbb{N}$ as follows:
$f(\frac{a,b}{)} = (a,b)$
This mapping is clearly onto, as every Cartesian pair can be mapped to a corresponding real number by taking $(a,b) \to \frac{a}{b}$
The cartesian product of $\mathbb{Z} \times \mathbb{N}$ is countably infinite
Therefore, the rational numbers are countably infinite.

□

Problem Q3. Show that the following problem is undecidable.

$$\mathbf{A} = \{< M > | \forall w \in \mathbf{S}\ M \text{accepts} w\} \text{ where } \mathbf{S} \text{ is the set of all strings}$$

Proof. Define a mapping reduction $R(< M >)$ as follows:

1. Erase the tape
2. Write ϵ to the tape
3. Run M on ϵ
4. Accept

Return $< M\# >$

If $< M > \in H_\epsilon$: M halts on ϵ, so $M\#$ accepts everything. Oracle $< M\# >$ accepts.
If $< M > \notin H_\epsilon$: M does not halt on ϵ, so $M\#$ accepts nothing and doesn't halt. Oracle $< M\# >$ rejects.
However, no machine to decide H can exist, so Oracle doesn't exist.

□

Problem Q4(a). Given n containers of weight $w_1, \ldots, w_n$, trucks of capacity K, minimize the number of trucks to carry all the weight. This problem is NP-Complete.
Greedy Algorithm: Start with an empty truck, then begin piling containers $1, 2, 3, \ldots$ until you get to a container which would overflow the weight limit.
Now declare this truck "loaded" and send it off; then continue the process with a fresh truck.
Given an example of a set of weights, a value of K, where this algorithm does not use the minimum possible number of trucks.

Proof. Let $K = 3$, $w_1, \ldots, w_{\frac{n}{2}} = 2, w_{\frac{n+2}{2}}, \ldots, w_n = 1$, $n \geq 4$, n is even
Trivially, the ideal arrangement would be to have each truck carry one container of weight 2 and one of weight 1 to fully utilize every truck.
However, the first truck will be loaded with weight $w_1 = 2$, then $w_2 = 2$ would overload it, so it will leave with only weight 2.

□

Problem Q4(b). Show, however, that the number of trucks used by this algorithm is within a factor of 2 of the minimum possible number, for any set of weights and any value K

Proof. Claim: The average utilization of each truck will not be below $\frac{1}{2}$
Suppose some truck currently carrying weight a cannot accept w_i, because $w_i + a > K$
This means $a > K - w_i$ by definition.
The minimum usage of this truck would therefore be $\frac{K - w_i + 1}{K}$
The next truck is guaranteed to use at least $b = w_i$ weight, since it is initially empty and w_i is the first container to be loaded.
Therefore, the average utilization of these two trucks is $\frac{1}{2}\left(\frac{K-w_i+1}{K} + \frac{w_i}{K}\right)$
$= \frac{K+1}{2K} > \frac{K}{2K} = \frac{1}{2}$
Extending this, the average usage of any arbitrary adjacent pair of trucks is guaranteed to be at least $\frac{1}{2}$
Now, given a container assignment, choose adjacent pairs such that each pair is disjoint from all other pairs.
Given N trucks used, this situation is equivalent to having N trucks each using precisely $\frac{1}{2}$ of their max capacity.
The lower bound for the minimum number of trucks N_m trivially occurs when all trucks use precisely

100% of their max capacity.
So given that each truck, in the worst situation with the greedy algorithm, use at least 50% of their max capacity, the number of trucks used by the greedy algorithm must be within a factor of 2 of the minimum possible number.

$\square$