Name: Kevina Wong
ID: 109170049

**CSCI 3104, Algorithms** **Profs. Chen & Grochow**
**Problem Set 11 – Due Wed April 29 11:55pm** **Spring 2020, CU-Boulder**

---

*Advice 1*: For every problem in this class, you must justify your answer: show how you arrived at it and why it is correct. If there are assumptions you need to make along the way, state those clearly.

*Advice 2*: Informal reasoning is typically insufficient for full credit. Instead, write a logical argument, in the style of a mathematical proof.

**Instructions for submitting your solutions**:

- All submissions must be typed.
- You should submit your work through the **class Canvas page** only.
- You may not need a full page for your solutions; pagebreaks are there to help Gradescope automatically find where each problem is. Even if you do not attempt every problem, please allot at least as many pages per problem (or subproblem) as are allotted in this template.

Quicklinks: 1 2

---

Name: $\boxed{\text{Kevina Wong}}$

ID: $\boxed{109170049}$

**CSCI 3104, Algorithms** **Profs. Chen & Grochow**

**Problem Set 11 – Due Wed April 29 11:55pm** **Spring 2020, CU-Boulder**

1. Indiana Jones is gathering $n$ artifacts from a tomb, which is about to crumble and needs to fit them into 5 cases. Each case can carry up to $W$ kilograms, where $W$ is fixed. Suppose the weight of artifact $i$ is the positive integer $w_i$. Indiana Jones needs to decide if he is able to pack all the artifacts. We formalize the Indiana Jones decision problem as follows.

   - Instance: The weights of our $n$ items, $w_1, \ldots, w_n > 0$.
   - Decision: Is there a way to place the $n$ items into different cases, such that each case is carrying weight at most $W$?

   Show that Indiana Jones $\in$ NP.

   **Answer:**
   Recall that an NP problem is defined as the set of decision problems such that if the answer is yes, then there exists a proof of that fact that can be checked in polynomial time.

   In this particular problem, we have $n$ artifacts, with weights $w_1, w_2, w_3 ... w_n$. We also have five cases, $\{c_1, c_2, c_3, c_4, c_5\} \in C$, with a fixed max kilograms, $W$. Let $m$ denote the cardinality of set $C$ (in this problem, it would be 5). Let $w(c_i)$ denote the total weight of the $i^{th}$ case (ie. $w(c_1)$ would denote the total weight of artifacts held by $c_1$). To prove that this problem is an NP problem, we must prove that a solution can be checked in polynomial time. There are two criteria for the problem to yield a "Yes" answer:

   (a) $n = 0$ because he wants to pack *all* the artifacts - the number of remaining artifacts would be 0.

   (b) For each $c_i \in C$, $w(c_i) \leq W$ because the weights of the cases cannot exceed the threshold.

   The certificate for this problem would be a "mapping" from each $n$ to a case $c_i \in C$ To check criteria (a), it would simply be $\mathcal{O}(1)$. To check criteria (b), it would be a run time of $\mathcal{O}(n)$ because for each of the $n$ artifacts, you would have compute the weight of its case, $w(c_i)$ and check $w(c_i) \leq W$ for all of $C$ which would take at most $\mathcal{O}(n)$ time. Therefore the total run time to check criteria (b) is $\mathcal{O}(n)$. Therefore, we can see that the runtime to check the solution would be in polynomial time. Thus, we can conclude that Indiana Jones $\in$ NP.

Name: Kevina Wong
ID: 109170049

**CSCI 3104, Algorithms** **Profs. Chen & Grochow**
**Problem Set 11 – Due Wed April 29 11:55pm** **Spring 2020, CU-Boulder**

2. A student has a decision problem $L$, which they know belongs to $\mathsf{NP}$. This student wishes to show that $L$ is $\mathsf{NP}$-$\mathsf{Complete}$. They attempt to do so by constructing a polynomial time reduction from $L$ to $\mathsf{SAT}$, a known $\mathsf{NP}$-$\mathsf{Complete}$ problem. That is, the student attempts to show that $L \leq_p \mathsf{SAT}$. Determine if this student's approach is correct and justify your answer.

   **Answer:**

   Recall the *Cook-Levin Theorem* that says the Boolean Satisfiability (SAT) problem is NP-complete. Also recall that to prove that a problem, $L$, is NP-complete, we can reduce a known NP-complete problem to $L$. Lastly, recall the following definitions:

   - **NP-Hard:** A problem, L, is NP-Hard if (a) for every $Q \in NP$ and (b) $Q \leq_p L$.
   - **NP-Complete** A language L is NP-Complete if (1) $L \in NP$ and (2) $L$ is NP-Hard.

   Overall, the concept is correct where the student must reduce $L$ to an already known NP-complete problem, SAT; However, they are incorrect at one part. Looking at the definitions above, we know criteria (1) for NP-Complete is already fulfilled as a given in the question. Criteria (2) for NP-complete says that $L$ must be NP-hard. We know that criteria (a) is satisfied from the Cook-Levin theorem stated above. If SAT $\in$ NP-complete, then SAT $\in$ NP because NP-complete $\in$ NP. Criteria (b) for NP-hard is where the student is incorrect. The student should **not** attempt to show $L \leq_p$ SAT; Rather, they **should** show SAT $\leq_p L$. To add to this explanation, the idea of *polynomial time reducibility* is that SAT and $L$ are problems such that SAT can be polynomial time reducible to $L$. Then, an algorithm used to solve SAT can be used to solve $L$. Here, the student flip-flopped the order to say that he/she will show that $L$ can be polynomial time reduced to SAT, which is incorrect and would not be useful as this would further imply that "existing" algorithms for $L$ can be used to solve SAT. This is backwards of what we actually want and would not successfully show NP-completeness of L.