1. **(12 pts) Short Answers.**

   (a) **(4 pts)** Suppose that Q is a decision problem. How do we express Q as a language?

   L is the set of strings x for which the answer to Q(x) is YES

   (b) **(4 pts)** How is the input given to a Turing machine? What is the size of the input?

   It is written on the tape.
   The length is the number of cells containing input characters.

   (c) **(4 pts)** State Church's Thesis.

   Anything that we would reasonably call a logical calculation can be modelled by a Turing machine.

2. **(5 pts)** The following algorithm tests whether a given integer can be expressed as the sum of three perfect squares:

Input: n (a positive integer)
for $x = 0$ to n
    for $y = 0$ to n
        for $z = 0$ to n
            if $x^2 + y^2 + z^2 = n$ then return YES
return NO

Is this a polynomial time algorithm? Explain your answer.

NO

The running time is $O(n3)$ which is polynomial in the VALUE of the input. It needs to be a polynomial in $\log n$, which is the SIZE of the input.

3. **(19 pts)** Consider the following language.

$$A = \{(< P >, k) | \text{there are fewer than 10 even integers } x \geq k \text{ such that } P(x) \text{ returns a value in the range } [x^2 + k + 10, x^2 + 2k + 10].\}$$

(a) **(15 pts)** Prove that A is not decidable.

Assume that A is decidable, and that a Turing machine M decides A.
Then the following Turing machine decides the Halting problem

---

Input $\langle P \rangle, y$

Create the following description of a Turing Machine:

$\langle B \rangle =$ " Input x

Run $U(\langle P \rangle, y)$ # U is the universal T. M.

return $x^2 + 13$ "

Run $M(\langle B \rangle, 2)$ to determine whether $(\langle B \rangle, 2) \in A$

If $(\langle B \rangle, 2) \in A$ then $\rightarrow q_R$ else $\rightarrow q_A$

We can create $\langle B \rangle$ by copying the input $\langle P \rangle, y$ into the appropriate part of the text.

Every step takes finite time, since M is a decider.

If $P(y)$ halts then $B(x)$ returns x^2+13 for every x. So it returns a number in the range $[x^2+12, x^2+14]$ for at least ten values of $x \geq 2$ and so $(\langle B \rangle, 2) \notin A$.

If $P(y)$ does not halt then $B(x)$ returns a value for Zero inputs x. Since $0 < 10$, $(\langle B \rangle, 2) \in A$.

So this TM returns the correct answer and decides the Halting problem. But the H.P. is undecidable. CONTRADICTION.

$\therefore$ A is not decidable.

(b) **(4 pts)** Either A or $\bar{A}$ is recognizable. Say which one you think is recognizable. You don't need to prove your answer. But write a few sentences explaining why you think it is the correct choice.

$\overline{A}$ is recognizable. I think this because recognizing $\overline{A}$ requires recognizing what P(x) returns for only 10 values of x. I think we should be able to find those 10 values using dovetailing.

4. **(16 pts)** Give **two proofs** that the following problem is in NP.
One proof must use a verification algorithm, and the other proof must use a nondeterministic Turing machine.

TRIPLE-SUM
Input: A list of positive integers $a_1, ..., a_n$.
Question: Can the list be partitioned into two parts A_1, A_2 such that the sum of the numbers in A_1 is exactly three times the sum of the numbers in A_2?

For example, if the input is 7,3,1,8,1 then the answer is YES ($A_1 = 7, 8$).
If the input is 5,4,4,7,5 then the answer is NO.

Verification algorithm:
The certificate will be a partition where both parts have the same sum

Input: A list $A = a_1, \ldots, a_n$; two other lists A_1, A_2

1) Check that A_1, A_2 is a partition of A
For $i=1$ to n, delete a_i from either A_1 or A_2 (there may be multiple copies of the integer a_i)
if a_i is not in A_1 or A_2 then reject.
If $A_1 \neq \emptyset$ or $A_2 \neq \emptyset$ then reject

2) Reset $A_1, A_2 :=$ the sets given as input
Compute the sum of the elements in A_1 and the sum of the elements in A_2

If $(\text{sum}(A_1) = 3\ \text{sum}(A_2))$ then ACCEPT
else REJECT.

Step 1 takes $O(n)$ time. Step 2 can be done in poly$(\lg a_1 + \lg a_2 + \ldots + \lg a_n)$ time. Both are polynomial in the size of the input file.

If TRIPLE-SUM(A) = YES then there is a partition A_1, A_2 with sum$(A_1) = 3$sum(A_2). The algorithm will accept $(A, (A_1, A_2))$.
If TRIPLE-SUM(A) = NO then there is no partition of A, where sum$(A_1) = 3 \cdot \text{sum}(A_2)$. So the algorithm will reject $(A, (A_1, A_2))$ for every A_1, A_2.

blank page

NDTM

Input $A = a_1, \ldots, a_n$

$A_1 = A_2 = \phi$

For $i=1$ to n

put a_i into either A_1 or A_2

compute the sum of A_1 and the sum of A_2

if $\text{sum}(A_1) = 3\ \text{sum}(A_2)$ then accept

else reject

The nondeterministic loop takes $O(n)$ steps

The two sums can be computed in $\text{poly}(\log a_1 + \log a_2 + \ldots + \log a_n)$ steps

Both are polynomial in the size of the input files

If TRIPLE-SUM(A) = YES then there is a partition A_1, A_2 with $\text{sum}(A_1) = 3\ \text{sum}(A_2)$. It is possible that the non-deterministic loop will choose this partition and so the algorithm will accept.

If TRIPLE-SUM(A) = NO, then no matter what partition is chosen, we will have $\text{sum}(A_1) \neq \text{sum}(A_2)$ and so the algorithm will reject.

5. **(12 pts)** Recall that the Halting Problem is:

Input: $< P >$, the description of an algorithm (i.e. Turing Machine), and a string X.
Question: Does P halt when given X as input?

Prove that there is no algorithm (i.e. Turing machine) that solves the Halting Problem.

Do not use a reduction; in other words, you cannot make use of the fact that some other language is not decidable.

Hint: You saw this proof in the second lecture.

Suppose that Turing machine M decides the Halting Problem.

Then we can use M to create the following TM

$\langle D \rangle$

the description of a Turing Machine

Input $\langle P \rangle$

decide whether $P(\langle P \rangle)$ halts

1) Run $M(\langle P \rangle, \langle P \rangle)$
2) If $M(\langle P \rangle, \langle P \rangle) \rightarrow$ accept then loop forever
3) If $M(\langle P \rangle, \langle P \rangle) \rightarrow$ reject then halt accept

$M(\langle P \rangle, \langle P \rangle)$ takes finite time since M is a decider

Consider running $D(\langle D \rangle)$

If $D(\langle D \rangle)$ halts then $M(\langle D \rangle, \langle D \rangle) \rightarrow$ accept so $D(\langle D \rangle)$ enters an infinite loop in line 2

If $D(\langle D \rangle)$ does not halt then $M(\langle D \rangle, \langle D \rangle) \rightarrow$ reject so $D(\langle D \rangle)$ halts in line 3

CONTRADICTION

$\therefore$ M does not exist, i.e. the H P is not decidable