1. (12 pts) Short Answers.

(a) (4 pts) Name any NP-complete problem.

   \textit{CNF-SAT}

(b) (4 pts) State Church's Thesis.

   Any logic-based algorithm can be encoded as a Turing Machine

(c) (4 pts) What does it mean for a verification algorithm to run in polynomial time?

   A verification algorithm takes two inputs: \( I, C \).

   To run in polytime, it must always run in at most \( f(\|I\|) \) steps, where \( f \) is a polynomial.
2. (10 pts) This figure represents a Turing Machine at some point during its run.

(a) (5 pts) Give the configuration that represents this figure.

(b) (5 pts) Part of the transition function says:

\[(q_3, y) \rightarrow (q_5, t, \text{Left})\]

Draw a figure representing the Turing Machine after the next step.
3. (30 pts) Consider the following language:

\[ A = \{ \langle P, k \rangle : \text{there are at least } k \text{ integers } x \text{ such that } P(x) \text{ halts and returns a number that is at least } xk + 5 \} \]

\( P \) is a Turing Machine with non-negative integers as inputs and outputs; \( k \) is a non-negative integer.

(a) (15 pts) Is \( A \) decidable? Prove your answer.

**NO**

Suppose that \( Q \) decides \( A \). Then the following algorithm will solve the Halting Problem:

Input: \( \langle H, y \rangle \)

Create the following text file:

\[ \langle P \rangle \\
\text{Input } x \\
\text{Run } U(\langle H, y \rangle) \\
\text{return } 3x + 5 \]

Run \( Q(\langle P, 3 \rangle) \) to determine whether \( (\langle P, 3 \rangle) \in A \)

If \( (\langle P, 3 \rangle) \notin A \) then accept \( \# \) indicating \( H(y) \) does not halt
If \( (\langle P, 3 \rangle) \in A \) then reject \( \# \) indicating \( H(y) \) halts

Note that to create the text file, we simply copy \( \langle H, x \rangle \) from the input into the appropriate place.

If \( H(y) \) does not halt, then for every \( x \), \( P(x) \) will not halt, so \( (\langle P, 3 \rangle) \notin A \)

If \( H(y) \) halts, then \( P(x) \) returns a value that is at least \( 3x + 5 \), so \( (\langle P, 3 \rangle) \in A \)

This decides the Halting Problem. But the Halting Problem is not decidable. **CONTRADICTION.**

A, \( \langle P, 3 \rangle \), \( H \) is a \( \langle \rangle \) which is possible.
(b) (15 pts) Either $A$ or $\overline{A}$ is recognizable. Which one? Prove that it is recognizable.

$A$ is recognizable. The following algorithm will recognize it:

Input $\langle P \rangle, k$

For $s = 1$ to $\infty$

$c := 0$

For $j = 1$ to $s$

Run $U(\langle P \rangle, j)$ for $s$ steps

# $U$ is the Universal Turing Machine

if $U(\langle P \rangle, j)$ halts within $s$ steps and
returns a value $\geq j \cdot k + 5$ then $c := c + 1$

if $c = k$ then halt accept

If this halts accept, then there were $k$ distinct values
of $j$ for which $P(j)$ halts and returns $\geq j \cdot k + 5$.

So $\langle \langle P \rangle, k \rangle \in A$

If $\langle P \rangle, k \in A$ then there are at least $k$ values $x_1, x_2, \ldots, x_k$ for which $P(x_i)$ returns $\geq x_i \cdot k + 5$. There is some integer $s^*$

that is greater than each of $x_1, x_2, \ldots, x_k$ and $s^*$ is

greater than the number of steps taken by $U(\langle P \rangle, x_1), \ldots, U(\langle P \rangle, x_k)$

This procedure will halt accept when $s = s^*$.
4. (24 pts) Consider the following problems.

**COST-OF-MAX-PATH**
**Input:** A connected graph $G$, with an integer weight $w_e \geq 0$ on each edge $e$, and two specified vertices $u, v$.
**Output:** The total cost of a path in $G$ from $u$ to $v$ with the greatest total weight.

Note: a path cannot repeat any vertices. So a maximum weight path can only visit each vertex at most once. The total weight of a path is the sum of the weights of all the edges on the path.

**DCMP**
**Input:** A connected graph $G$, with an integer weight $w_e \geq 0$ on each edge $e$, two specified vertices $u, v$, and an integer $T \geq 0$.
**Question:** Is there a path in $G$ from $u$ to $v$ of total weight at least $T$?

DCMP stands for Decision-Cost-of-Max-Path.

(a) (6 pts) Prove that DCMP is in NP.

This is a decision problem.
If the answer is YES, then the certificate is
a sequence of vertices $V_0 = (x_0, x_1, x_2, \ldots, x_e = V$ which form a path of weight $\geq T$.

The verification algorithm confirms:
(i) $(x_i, x_{i+1})$ is an edge for each $i$;
(ii) the weights of the edges total $\geq T$

Part (i) takes $O(n)$ time (on an $n$-vertex graph).
The time required for (ii) is proportional to the total # of digits in the weights on the path which is $\leq$ the size of the input file containing those weights.

So this is a poly time verification algorithm.
(b) (8 pts) Show that if there is a polytime algorithm for DCMP then there is a polytime algorithm for COST-OF-MAX-PATH.

Let $W$ be the total of all the weights.
The # of digits in $W$ is $\leq$ the total # of digits of all weights which is $\leq$ the size of the input file.

So $\log W$ is $O(\text{input file})$

Every path in $G$ has total weight $\leq W$

If $Q$ solves DCMP in poly-time, then we can use $Q$, along with binary search, to find the largest $0 \leq T \leq W$ such that DCMP($G$, $u$, $v$, $T$) returns YES. $T$ is the correct answer for COST-OF-MAX-PATH ($G$, $u$, $v$).

Binary search requires $\log W$ calls to $Q$.

Since $Q$ and $\log W$ are both polynomial in the size of the input, this is a polytime algorithm.
(c) (10 pts) Recall the problem:

**HAM-PATH**

**Input:** A connected graph $G$ and two specified vertices $u, v$.

**Question:** Is there a Hamilton path in $G$ from $u$ to $v$?

Recall that a Hamilton path visits every vertex in $G$ exactly once. (Usually we don’t specify that $G$ is connected, but it is convenient to do so here, and it makes little difference since the condition is easy to check.)

Prove that $\text{DCMP} \geq_P \text{HAM-PATH}$

$G, u, v$ is an input to $\text{HAM-PATH}$

Set $n$ = the # of vertices in $G$.

We create an input for $\text{DCMP}$ as follows:

a) Give every edge weight $w_e = 1$

b) Set $T = n - 1$

A path $\pi$ from $u$ to $v$ has weight $\geq 2$ iff it contains at least $n - 1$ edges; i.e., iff it is a Hamilton path.

$\therefore \text{DCMP}(G, u, v, T)$ has the same answer as $\text{HAM-PATH}(G, u, v)$

Take 1 step for each edge

b) Takes 1 step

The trans for motion takes poly time

$\therefore \text{DCMP} \geq_P \text{HAM-PATH}$
5. (8 pts) A, B, C are recognizable languages. Prove that \( A \cup B \cup C \) is recognizable.

\[
\begin{align*}
\text{Suppose } Q_A & \text{ recognizes } A \\
Q_B & \text{ recognizes } B \\
Q_C & \text{ recognizes } C \\
\text{The following algorithm will recognize } A \cup B \cup C \\
\end{align*}
\]

\[
\begin{align*}
\text{Input } x \\
\text{for } s = 1 \text{ to } \infty \\
\text{run } Q_A(x) \text{ for } s \text{ steps} \\
\text{run } Q_B(x) \text{ for } s \text{ steps} \\
\text{run } Q_C(x) \text{ for } s \text{ steps} \\
\text{if either one halts accept, then accept.}
\end{align*}
\]

If \( x \in A \cup B \cup C \) then neither of \( Q_A(x), Q_B(x), Q_C(x) \) will halt accept, so this algorithm will not halt accept.

If \( x \in A \cup B \cup C \) then at least one of \( Q_A(x), Q_B(x), Q_C(x) \) will halt accept. Let \( s_x \) be the number of steps it takes. Then this algorithm will accept at iteration \( s = s_x \).

\[\therefore \text{This algorithm recognizes } A \cup B \cup C\]