Problem 1

Solution. Assume the PDA that recognizes $A$ is $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$. We construct another PDA $M'$ that recognizes $\text{PREFIX}(A)$ based on $M$.

For each vertex $v$ in the diagram of $M$, we do the following construction. For every path $P$ from $v$ to an accept state, we put an replication $P'$ of $P$ on the diagram. Here $P'$ starts at $v$ while all its edges and vertices (except $v$) are replications of the corresponding ones in $P$, newly added to the diagram. In particular, states in $P'$ corresponding to accept states in $P$ will also be accept states.

Next, for each newly added path $P'$, we do the following modification. For every edge on $P'$, assume the corresponding edge on $P$ has the transition rule $(q, b) \in \delta(p, c, a)$ for some $p, q \in Q, c \in \Sigma_c, a, b \in \Gamma_c$. Then we change the transition rule on this edge to be $(q', b) \in \delta(p', c, a)$, where $p', q'$ are the new states corresponding to $p, q$. That is, from this point on $M'$ is going to transit just according to the stack data and ignore the input.

Now we prove that $M'$ recognizes $\text{PREFIX}(A)$.

For every string $s \in \text{PREFIX}(A)$, according to the definition of $\text{PREFIX}(A)$, there exists $t \in \Sigma^*$, such that $w = st \in A$. If $M$ reaches the vertex $u$ by reading $s$, having stack data $S$, we know that from $u$, by reading $t$, $M$ reaches an accept state through a path $P$. So by reading $s$, $M'$ can also reach the state $u$, having the same stack data. According to our construction, there is a path $P'$, which is a modified replication of $P$, which starts at $u$ and ends at an accept state. According to our settings for the transition rule, we know that $M'$ can reach an accept state from $u$, by reading $c$. So $s$ is accepted by $M'$.

For every string $s$ accepted by $M'$, by reading $s$, if $M'$ reaches an accept state by following a path in the diagram of $M$, then $s$ is in $A$. On the other hand, consider the case that $M'$ reaches an accept by first following a path in $M$ to reach some vertex $u$ and then following a modified replicated path $P'$. Assume $P'$ is a modified replication of $P$ in $M$. According to our construction, following $P'$, $M'$ cannot read any character in $\Sigma$. So $M'$ must reach $u$ by reading exactly $s$. On the other hand, as $P$ and $P'$ have the same stack operations, there exists a string $t$ corresponding to $P$ such that by reading $t$, $M$ reaches an accept state from $u$, following the path $P$. So $st$ is accepted by $M$. So $s \in \text{PREFIX}(A)$. □

Problem 2

Solution. Assume the pumping length of $L$ is $p$. Consider the string $s = 0^{2^p+1}$. We know $2^{p+1} \geq p$ for every $p \in \mathbb{N}$. By the Pumping Lemma, $s$ can be written as $uvxyz$ where $|vy| > 0$, $|vxy| \leq p$, for every $i \in \mathbb{N}, uv^ixy^iz \in L$. Consider $s' = uv^0xy^0z$. We know that $|s'| \geq 2^{p+1} - p$. However, as $p < 2^p$ for any $p \in \mathbb{N}^+$, $2^{p+1} - p > 2^p$. This shows that $s' \notin L$ which is a contradiction. □

Problem 3

Solution. Assume the pumping length of $B$ is $p$. Consider the string $s = 1^p0^{2p}1^p$. By the Pumping Lemma, $s$ can be written as $uvxyz$ where $|vy| > 0$, $|vxy| \leq p$, for every $i \in \mathbb{N}, uv^ixy^iz \in L$.

Notice that there are only these 4 possible cases of $vy$, because $|vxy| \leq p, |vy| > 0$ and the number of 0s in $s$ is $2p \geq p$.

- If $vy = 0^{c_1}$ for some $c_1 \leq p$, then $uv^2xy^2z$ have more 0s than 1s.
• If $vy = 1^{c_2}$ for some $c_2 \leq p$, then $uv^2xy^2z$ have more 1s than 0s.

• If $vy = 0^{c_3}1^{c_4}$ for some $c_3, c_4 \leq p$, then $uv^0xy^0z = 1^p0^a1^b$ is not a palindrome, as $b < p$.

• If $vy = 1^{c_5}0^{c_6}$ for some $c_5, c_6 \leq p$, then $uv^0xy^0z = 1^c0^d1^p$ is not a palindrome, as $c < p$.

So this is a contradiction.

\[\square\]

**Problem 4**

**Solution.** Assume the two-dimensional TM is $M$. It runs in time $T(n)$. As $M$ starts at $(0,0)$ on the two-dimensional tape, all positions that $M$ can visit on the tape are in $[T(n)] \times [T(n)]$ where $[T(n)] = \{0,1,\ldots,T(n)\}$. Next we construct a 1-dimension TM $M_1$.

Let $M_1$ have 2 tapes, $t_1$ and $t_2$.

We map the two-dimensional tape of $M$ to the tape $t_1$ of $M_1$. For any position $(p, q)$ on the 2-dimension tape, we map $(p, q)$ to the position $x$ on tape $t_1$ in the same way as we map rational numbers to natural numbers in our class. Assume $(p, q)$ is on the $d$th diagonal of the $[T(n)] \times [T(n)]$ square. Here the $d$th diagonal is defined to be the position series $(0, d - 1), (1, d - 2), \ldots, (d - 1, 0)$. The number of positions on the diagonal is $d = p + q + 1$. Let $x = \sum_{i=1}^{d-1} + p$.

The tape $t_2$ is used to record the current position of the r/w head $s$ of $M$, a number (counter) $c$ which indicates how many steps are left to move $s_1$ from its current position to its destination $x$. Also $t_2$ is used as a working tape for some intermediate computations.

$M_1$ simulates one step of $M$ as follows. First $M_1$ updates the symbol at the current position, and computes the next position $(p, q)$ of head $s$ of $M$ according to the transition function of $M$. Then $M_1$ computes the corresponding position $x$ of $(p, q)$. After that $M_1$ computes the distance $c$ between $x$ and the current position of $s_1$. Then $M_1$ starts to move $s_1$ to its destination. Each time $s_1$ moving 1 step, $M_1$ decreases $c$ by 1. $M_1$ keeps moving $s_1$ until $c = 0$. As $c = O(T(n))$, one move on the 2-dimension tape corresponds to at most $O(T(n))$ steps on the 1-dimension tape $t_1$. Since $M$ runs in time $O(T(n))$, $M_1$ runs in time $O(T(n)^2)$.

\[\square\]

**Remark 1.** If you put the rows or columns of the two-dimensional tape sequentially on the 1-dimensional tape, then the simulation time is $O(T(n)^3)$. The reason is that $T(n)$ is not known in advance, and therefore if one uses this approach there can be times when an insertion in previous tapes will result in the moving of all later symbols to the right, which can take time $O(T(n)^2)$. Therefore simulating one step of the 2-grid TM will take time $O(T(n)^2)$ and the total time is $O(T(n)^3)$.