Theorem

Let $\Sigma = \{0, 1, \#\}$.
The set of Palindromes

$$PAL = \{z \epsilon \ \Sigma^* \mid z = revz\}$$

requires $\Omega(n^2)$ time on a single tape Turing Machine.

Proof :

$PAL_n = \{$ x $\#^{n/2}$ rev x $| \, x \epsilon$ $\{0,1\}^{n/4}\}$
where n is a multiple of 4.

$$PAL_n \subseteq PAL$$

For each $x \; \epsilon \; PAL_n$ and for each position i, $0 \leq i \leq n$, let $c_i(x)$ denote the sequence $q_1, q_2, q_3, ..., q_k$ of states of the finite control Q of M that M is in as it passes over the line between the i^{th} symbol and $i + 1^{st}$ symbol in either direction while scanning x.

$c_i(x)$ is called the Crossing Sequence at i.

Let

$$C(x) = \{c_i(x) \mid n/4 \leq i \leq 3n/4\}$$

Lemma: If $x, y \in$ PAL$_n$ and x $\neq$ y, then

$$C(x) \cap C(y) = \emptyset \ .$$

Proof : Suppose c $= C(x) \cap C(y)$. Let c $=$ c$_i(x) =$ c$_j(y)$ Let x' be the prefix of x consisting of the first i symbols and y' be the suffix of y consisting of the last $n - j$ symbols.

Then $x'y'$ will be accepted by M.

But $x'y'$ is not in PAL$_n$, since it is not a palindrome.

This is a contradiction
Therefore

$$C(x) \cap C(y) = \emptyset.$$

Let m_x be the length of the smallest crossing sequence in $C(x)$.

Let $m = max\{m_x \mid x \epsilon PAL_n\}$

Number of Crossing Sequences
of length atmost $m = \sum_{i=1}^{m} \mid Q \mid^i = \frac{|Q|^{m+1}-1}{|Q|-1}$

Number of elements of $PAL_n = 2^{n/4}$

Since all the shortest crossing sequences of PAL$_n$ must be different,

$$2^{n/4} \leq \sum_{i=1}^{m} \mid Q \mid^i = \frac{|Q|^{m+1}-1}{|Q|-1}$$

We get

$$m \geq \Omega(n)$$

Since m_x is the length of the shortest crossing sequence in C(x), all crossing sequences in PAL_n are of length $\geq \Omega(n)$.

Therefore, it takes at least $n/2.\Omega(n) = \Omega(n^2)$ time to generate all the crossing sequences in C(x).

Theorem

A Turing Machine that accepts a Non-regular set uses at least $\Omega(loglogn)$ space.

M has a read only input tape and a read/write work tape,and that M always moves its input head all the way to the right of the input string before accepting.

If M is s(n) space bounded , Number of possible configurations,

$$N = q.s(n).d^{s(n)}$$

where q is the number of states and d is the size of the worktape alphabet of M.

Taking logarithm on both sides, we get

$$\log N = \log q + \log s(n) + s(n) \log (d)$$

Since q,d are independent of s(n), we can say,

$$s(n) = \Omega(logN) \quad (1)$$

In this proof, the crossing sequence at i consist of sequence of such configurations occuring at position i in the input string in either direction.

Number of possible Crossing Sequences of length atmost m $= \sum_{i=1}^{m} \mid N \mid^i = \frac{|N|^{m+1}-1}{|N|-1}$

Lemma : If there is a fixed finite bound k on the amount of space used by M on accepted inputs, then L(M) is a regular set.

Proof : We can modify M to mark off k cells initially. (k can be kept in the finite control) Whenever computation tries to use more than k cells, we reject it.

But then, no worktape memory would be required at all. All contents of the worktape can be kept in the finite control and then, M is equivalent to a two way finite automaton.

Since L(M) is not regular, for each k, there exist a string, x ϵ L(M) of minimum length for which at least k worktape cells are used.

There are n/2 distinct crossing sequences

In order to have n/2 distinct crossing sequences, there must be a crossing sequence of length at least m, where

$$n/2 \leq \sum_{i=1}^{m} \mid N \mid^{i} = \frac{|N|^{m+1}-1}{|N|-1}$$

Taking logarithm on both sides, we get

$$m = \Omega(logn) \tag{2}$$

If a crossing sequence is of length $\geq$ 2N, it would mean a configuration would appear in the crossing sequence twice in the same direction, which implies that M is looping.

$$m \leq 2N$$

we can state,

$$N = \Omega(m) \tag{3}$$

From (1) and (3), we get

$$s(n) = \Omega(logm) \tag{4}$$

From (4) and (2), we get

$$s(n) = \Omega(loglogn) \tag{5}$$

which is the required result.