The Complexity of Node Counting on Undirected Graphs

Boleslaw Szymanski
Department of Computer Science, Rensselaer Polytechnic Institute
szymansk@cs.rpi.edu

Sven Koenig
College of Computing, Georgia Institute of Technology
skoenig@cc.gatech.edu

Abstract

We analyze the complexity of Node Counting, a graph-traversal method. We show that the complexity of Node Counting on undirected graphs is $\Omega\left(n^{\sqrt{(1/6-\epsilon)n}}\right)$, where $0 < \epsilon < 1/6$ is an arbitrarily small constant and n is the number of vertices.

1 Introduction

Node Counting is a simple graph-traversal method that has been used in artificial intelligence to explore unknown environments, either on its own or to accelerate reinforcement-learning methods. To the best of our knowledge, the term "Node Counting" was first used in [Thrun, 1992]. Later, it has been suggested that variants of Node Counting approximate the exploration behavior of ants, that use pheromone traces to guide their exploration [Wagner *et al.*, 1997]. Node Counting is also similar to "Avoiding the Past: A Simple but Effective Strategy for Reactive Navigation" [Balch and Arkin, 1993] with mobile robots. In this paper, we present an undirected tree on which, perhaps surprisingly, the complexity of Node Counting is $\Omega\left(n^{\sqrt{(1/6-\epsilon)n}}\right)$, where $0 < \epsilon < \frac{1}{6}$ is an arbitrarily small constant and n is the number of states (vertices). Hence, its complexity is not polynomial in the number of states.

We use the following notation to describe Node Counting: S denotes the finite set of states of the domain, $s_{start} \in S$ the start state, and $\emptyset \neq G \subseteq S$ the set of goal states. The number of states is $n := |S|$. $A(s) \neq \emptyset$ is the finite, nonempty set of actions that can be executed

Initially, the u-values $u(s)$ are zero for all $s \in S$.

1. $s := s_{start}$.
2. If $s \in G$, then stop successfully.
3. $a :=$ one-of arg $\min_{a \in A(s)} u(succ(s,a))$.
4. $u(s) := 1 + u(s)$.
5. Execute action a. This changes the current state to $succ(s,a)$.
6. $s :=$ the current state.
7. Go to 2.

Figure 1: Node Counting

in state $s \in S$. $succ(s,a)$ denotes the successor state that results from the execution of action $a \in A(s)$ in state $s \in S$. We measure the complexity of Node Counting in action executions and assume that one can reach a goal state from every state that can be reached from the start state. Domains with this property guarantee that Node Counting reaches a goal state eventually. We also use two operators with the following semantics: Given a set X, the expression "one-of X" returns an element of X according to an arbitrary selection rule. A subsequent invocation of "one-of X" can return the same or a different element. The expression "arg $\min_{x \in X} f(x)$" returns the elements $x \in X$ that minimize $f(x)$, that is, the set $\{x \in X | f(x) = \min_{x' \in X} f(x')\}$.

Node Counting is shown in Figure 1. A u-value $u(s)$ corresponds to the number of times Node Counting has already been in state s. Node Counting always moves to a successor state with a minimal u-value because it wants to get to states which it has visited a smaller number of times to eventually reach a state that it has not yet visited at all, that is, a potential goal state.

2 Complexity of Node Counting

In the following, we present an undirected tree that shows that the complexity of Node Counting on undirected graphs is $\Omega\left(n^{\sqrt{(1/6-\epsilon)n}}\right)$, where $0 < \epsilon < \frac{1}{6}$ is an arbitrarily small constant and n is the number of states.

Consider undirected trees of the kind shown in Figure 2. The trees have $m+1 \geq 3$ levels. The levels consist of states of three different kinds: g-subroots, r-subroots, and leaves that are connected to the subroots. g-subroots and r-subroots alternate. At level $i = 0$, there is one subroot, namely a g-subroot g_0. At levels $i = 1 \ldots m$, there are two subroots, namely an r-subroot r_i and a g-subroot g_i. Subroot g_i has $m+i$ leaves connected to it, and subroot r_i has one leaf connected to it. Finally, subroot g_m is connected to two additional states, namely the start state and the only goal state. The trees have $n = \frac{3}{2}m^2 + \frac{9}{2}m + 3$ states.

Node Counting proceeds in a series of passes through the tree. Each pass traverses the subroots in the opposite order than the previous pass. We call a pass that traverses the subroots in descending order a down pass, and a pass that traverses them in ascending order an up pass. We number passes from zero on upward. Thus, even passes are down passes and odd passes are up passes. A pass ends immediately before it changes directions.

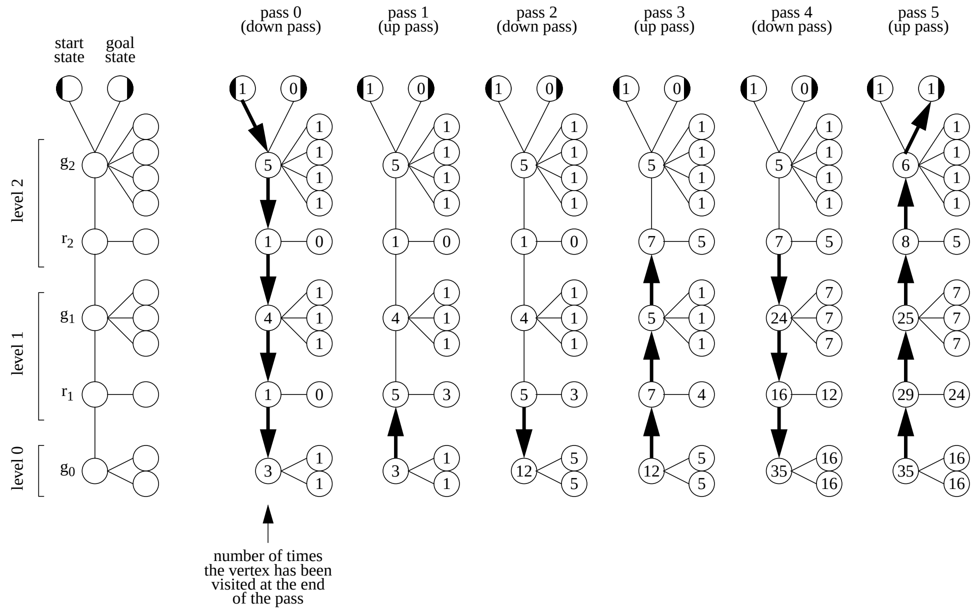

Figure 2: Node Counting has Exponential Runtime ($m = 2, n = 18$)

The semantics of the "one-of" operator on Line 3 in Figure 1 allows for the selection of any element of the operator's argument set. Hence, we present a selection rule for this operator that results in a bad performance. We make the selections as follows:

During pass zero, whenever possible, a leaf state is selected at g-subroots and a g-subroot is selected at r-subroots. The goal state is not selected. The selection of the subroot is unique. In case of leaves, any eligible leaf can be chosen. Pass zero ends in subroot g_0 after each leaf of subroot g_0 has been visited once. As a result, at the end of pass zero each subroot g_i has been visited $m + i + 1$ times, and each of its leaves has been visited once. Each r-subroot has been visited once, and its leaf has not been visited at all.

During all subsequent passes, whenever possible, a subroot is selected. If two r-subroots are eligible for selection (when Node Counting is at a g-subroot) then that r-subroot is chosen which extends the current pass. If two g-subroots are eligible (when Node Counting is at an r-subroot), the g-subroot is chosen in such a way as to terminate the current pass and start a new one in the opposite direction. When only leaves are eligible for selection, one of them is chosen arbitrarily.

From now on, we consider a tree with an arbitrary but constant number of levels $m+1 \geq 3$. Hence, in the three functions defined below, the argument m is omitted for the sake of brevity. Let $v_p(s)$ denote the total number of times that subroot s of the tree has been entered at the end of pass p. Our selection rules ensure that all leaves of a subroot have been entered the same number of times at the end of each pass, so we use $w_p(s)$ to denote the number of times each of the leaves of subroot s has been entered at the end of pass p. Finally, let $x_p(s)$ denote the total number of times subroot s has been entered from non-leaves at the end of pass p. By definition, $v_p(s)$, $w_p(s)$, and $x_p(s)$ are nondecreasing functions of p. These values relate as follows: The total number of times that a subroot has been entered at the end of pass p is equal to the product of the number of its leaves and the total number of times that it has been entered from each of its leaves at the end of pass p (which equals the total number of times that each of its leaves has been entered at the end of pass p) plus the total number of times the subroot has been entered from non-leaves at the end of pass p. For example, $v_p(g_i) = (m+i)w_p(g_i) + x_p(g_i)$.

Lemma 1 *Assume that Node Counting visits subroot s (with $s \neq g_m$) during pass p, where $0 < p < 2m+2$. The values $v_p(s)$ can then be calculated as follows:*

$$
\begin{aligned}
v_{2k+1}(g_0) = v_{2k}(g_0) &= m\, v_{2k}(r_1) + x_{2k}(g_0) \\
\text{for } 0 < i < m: & \\
v_{2k}(g_i) &= (m+i)\min(v_{2k-1}(r_i), v_{2k}(r_{i+1})) + x_{2k}(g_i) \\
v_{2k+1}(g_i) &= (m+i)\min(v_{2k}(r_{i+1}), v_{2k+1}(r_i)) + x_{2k+1}(g_i) \\
\text{for } 0 < i \leq m: & \\
v_{2k}(r_i) &= \min(v_{2k-1}(g_{i-1}), v_{2k}(g_i)) + x_{2k}(r_i) \\
v_{2k+1}(r_i) &= \min(v_{2k}(g_i), v_{2k+1}(g_{i-1})) + x_{2k+1}(r_i)
\end{aligned}
$$

Proof: Assume that Node Counting visits subroot g_0 during down pass $0 < 2k < 2m+2$. For $k = 1$, it holds that $w_1(g_0) = w_0(g_0) = v_0(r_1) \leq v_2(r_1)$. As long as the number of visits to any of the leaves of g_0 is less than $v_2(r_1)$, Node Counting moves to such a leaf. Thus, when Node Counting moves to r_1, it must be that $w_2(g_0) = v_2(r_1)$. For $k > 1$, it holds that $w_{2k-1}(g_0) = w_{2k-2}(g_0) = v_{2k-2}(r_1) \leq v_{2k}(r_1)$. Again, as long as the number of visits to any of the leaves of g_0 is less than $v_{2k}(r_1)$, Node Counting moves to such a leaf. Thus, when Node Counting moves to r_1, it must be that $w_{2k}(g_0) = v_{2k}(r_1)$. Hence, it holds in both cases that $v_{2k+1}(g_0) = v_{2k}(g_0) = mw_{2k}(g_0) + x_{2k}(g_0) = mv_{2k}(r_1) + x_{2k}(g_0)$.

Assume now that Node Counting visits subroot g_i (with $0 < i < m$) during down pass $0 < 2k < 2m+2$. As long as the number of visits to any of the leaves of subroot g_i is less than $\min(v_{2k-1}(r_i), v_{2k}(r_{i+1}))$, Node Counting moves to such a leaf. Hence, when Node Counting moves to another subroot, it must hold that $w_{2k}(g_i) = \max(\min(v_{2k-1}(r_i), v_{2k}(r_{i+1})), w_{2k-1}(g_i))$. Likewise, assume that Node Counting visits subroot g_i (with $0 < i < m$) during up pass $0 < 2k+1 < 2m+2$. Then, according to our selection rules, it holds that $w_{2k+1}(g_i) = \max(\min(v_{2k}(r_{i+1}), v_{2k+1}(r_i)), w_{2k}(g_i))$.

We now show by induction on p that, for all passes p (with $0 \leq p < 2m+2$) and all subroots g_i (with $0 < i < m$), it holds that $w_p(g_i) \leq \min(v_p(r_i), v_p(r_{i+1}))$. For pass zero, it holds that $w_0(g_i) = 1 \leq \min(1,1) = \min(v_p(r_i), v_p(r_{i+1}))$. Assume that the inequality holds for down pass $2k$. Then, if Node Counting visits subroot g_i during up pass $2k+1$, it holds that

$$
\begin{aligned}
w_{2k+1}(g_i) &= \max(\min(v_{2k}(r_{i+1}), v_{2k+1}(r_i)), w_{2k}(g_i)) \\
&\leq \max(\min(v_{2k+1}(r_i), v_{2k+1}(r_{i+1})), \min(v_{2k}(r_i), v_{2k}(r_{i+1}))) \\
&= \min(v_{2k+1}(r_i), v_{2k+1}(r_{i+1})),
\end{aligned}
$$

because $v_p(s)$ is a non-decreasing function of pass number p. If Node Counting does not visit subroot g_i during up pass $2k+1$, then it holds that $w_{2k+1}(g_i) = w_{2k}(g_i) \leq \min(v_{2k}(r_i), v_{2k}(r_{i+1})) \leq \min(v_{2k+1}(r_i), v_{2k+1}(r_{i+1}))$, again because $v_p(s)$ is a non-decreasing function of pass number p. Assuming that the inequality holds for up pass $2k+1$, a similar argument shows that the inequality continues to hold for the following down pass.

Now assume again that Node Counting visits subroot g_i (with $0 < i < m$) during down pass $0 < 2k < 2m + 2$. Then, as we just showed $w_{2k-1}(g_i) \leq \min(v_{2k-1}(r_i), v_{2k-1}(r_{i+1})) \leq \min(v_{2k-1}(r_i), v_{2k}(r_{i+1}))$ because $v_p(s)$ is a nondecreasing function of pass number p. Hence, it holds that

$$
\begin{aligned}
w_{2k}(g_i) &= \max(\min(v_{2k-1}(r_i), v_{2k}(r_{i+1})), w_{2k-1}(g_i)) \\
&= \min(v_{2k-1}(r_i), v_{2k}(r_{i+1})) \\
&\text{and finally} \\
v_{2k}(g_i) &= (m+i)w_{2k}(g_i) + x_{2k}(g_i) \\
&= (m+i)\min(v_{2k-1}(r_i), v_{2k}(r_{i+1})) + x_{2k}(g_i).
\end{aligned}
$$

A similar argument shows that it holds that $v_{2k+1}(g_i) = (m+i)w_{2k+1}(g_i) + x_{2k+1}(g_i) = (m+i)\min(v_{2k}(r_{i+1}), v_{2k+1}(r_i)) + x_{2k+1}(g_i)$.

The remaining two equalities defining the number of visits at r-subroots can be derived similarly. ■

We now use the lemma to prove the following theorem.

Theorem 1 *If $p = 2k$ for $0 \leq k \leq m$, then the down pass ends at subroot g_0 and it holds that*

$$
\begin{aligned}
v_{2k}(g_i) &= \begin{cases} m(v_{2k-2}(g_i) + 2k) + k + 1 & \text{for } i = 0 < k \\ (m+i)(v_{2k-2}(g_i) + 2k - 2i + 1) + 2k - 2i + 1 & \text{for } 0 < i < k \\ m + i + 1 & \text{otherwise} \end{cases} \\
v_{2k}(r_i) &= \begin{cases} v_{2k-1}(g_{i-1}) + 2k - 2i + 2 & \text{for } 0 < i \leq k \\ 1 & \text{otherwise} \end{cases} \\
x_{2k}(g_i) &= \begin{cases} k + 1 & \text{for } i = 0 \\ 2k - 2i + 1 & \text{for } 0 < i < k \\ 1 & \text{otherwise} \end{cases} \\
x_{2k}(r_i) &= \begin{cases} 2k - 2i + 2 & \text{for } 0 < i \leq k \\ 1 & \text{otherwise} \end{cases} \\
v_{2k}(r_i) &\geq v_{2k-1}(r_{i-1}) \qquad \text{for } 1 < i \leq k \\
v_{2k}(g_i) &> v_{2k-1}(g_{i-1}) \qquad \text{for } 0 < i < k
\end{aligned}
$$

If $p = 2k + 1$ for $0 \leq k \leq m$, then the up pass ends at subroot r_{k+1} (with the exception of up pass $2m + 1$, that ends at the goal state) and it holds that

$$
v_{2k+1}(g_i) = \begin{cases} v_{2k}(g_i) & \text{for } i = 0 \\ (m+i)(v_{2k-1}(g_i) + 2k - 2i) + 2k - 2i + 2 & \text{for } 0 < i < k \\ m + i + 2 & \text{for } 0 < i = k \\ m + i + 1 & \text{otherwise} \end{cases}
$$

$$
v_{2k+1}(r_i) = \begin{cases} v_{2k}(g_i) + 2k - 2i + 3 & \text{for } 0 < i \leq k \\ v_{2k+1}(g_{i-1}) + 2 & \text{for } i = k + 1 \leq m \\ 1 & \text{otherwise} \end{cases}
$$

$$
x_{2k+1}(g_i) = \begin{cases} k + 1 & \text{for } i = 0 \\ 2k - 2i + 2 & \text{for } 0 < i \leq k \\ 1 & \text{otherwise} \end{cases}
$$

$$
x_{2k+1}(r_i) = \begin{cases} 2k - 2i + 3 & \text{for } 0 < i \leq k \\ 2 & \text{for } i = k + 1 \leq m \\ 1 & \text{otherwise} \end{cases}
$$

$$
\begin{array}{lcll} v_{2k+1}(r_i) & \geq & v_{2k}(r_{i+1}) & \text{for } 0 < i \leq k \text{ and } i < m \\ v_{2k+1}(g_i) & > & v_{2k}(g_{i+1}) & \text{for } 0 \leq i < k \end{array}
$$

Proof by induction on the number of executed actions:

Part 1: The values are correct for $p = 0$.

$$
\begin{array}{lcll} v_0(g_i) & = & m + i + 1 & \text{for } 0 \leq i \leq m \\ v_0(r_i) & = & 1 & \text{for } 0 < i \leq m \\ & & & \\ x_0(g_i) & = & 1 & \text{for } 0 \leq i \leq m \\ x_0(r_i) & = & 1 & \text{for } 0 < i \leq m \end{array}
$$

At the end of the down pass, Node Counting is at subroot g_0 and is about to move to subroot r_1, starting an up pass.

For the remainder of the proof, notice that Node Counting cannot return to a subroot during a pass after it has moved from the subroot to a different subroot (such a return constitutes a change of direction, so it ends the current pass and starts a new one).

Part 2: Assume that $p = 2k + 1$ for $0 \leq k \leq m$. Up pass $2k + 1$ starts where the previous down pass ended, that is, at subroot g_0. We distinguish the following six cases:

1: $s = g_0$ for $0 \leq k \leq m$.

Value $x_{2k+1}(g_0)$: According to the induction hypothesis, it holds that $x_{2k+1}(g_0) = x_{2k}(g_0) = k + 1$.

Value $v_{2k+1}(g_0)$: According to the induction hypothesis, Node Counting starts the up pass at subroot g_0 and the next subroot that Node Counting visits is r_1. Thus, it holds that $v_{2k+1}(g_0) = v_{2k}(g_0)$.

Inequality $v_{2k+1}(g_i) > v_{2k}(g_{i+1})$ for $0 = i < k$: According to the induction hypothesis, it holds for $0 \leq l \leq k$ that

$$
v_{2l+1}(g_0) = v_{2l}(g_0) = \begin{cases} m + 1 & \text{for } l = 0 \\ m(v_{2l-2}(g_0) + 2l) + l + 1 & \text{otherwise.} \end{cases}
$$

From this definition, it follows that $v_{2l+1}(g_0) > mv_{2l-1}(g_0)$ for $l > 0$ and by induction on l we get $v_{2l+1}(g_0) > v_1(g_0)m^l = (m+1)m^l$ for $l > 0$. Solving the recursion yields

$$v_{2l+1}(g_0) = v_{2l}(g_0) = \frac{m^{l+3} + m^{l+2} + m^{l+1} - (2l+2)m^2 + (l-2)m + l + 1}{m^2 - 2m + 1} \quad (1)$$

Similarly, according to the induction hypothesis, it holds for $1 \leq l \leq k$ that

$$v_{2l}(g_1) = \begin{cases} m+2 & \text{for } l = 1 \\ (m+1)(v_{2l-2}(g_1) + 2l - 1) + 2l - 1 & \text{otherwise.} \end{cases}$$

Solving the recursion yields

$$v_{2l}(g_1) = \frac{(m+1)^{l-1}(m^3 + 5m^2 + 8m + 4) - (m^2 + 4 + 2m^2 l + 4ml + 4m)}{m^2}.$$

Using the previous results, we verify for $2 \leq m < 5$ and $0 < k \leq m$ that $v_{2k+1}(g_0) > v_{2k}(g_1)$ (see the table below).

m	$v_3(g_0)$	$v_2(g_1)$	$v_5(g_0)$	$v_4(g_1)$	$v_7(g_0)$	$v_6(g_1)$	$v_9(g_0)$	$v_8(g_1)$
2	12	4	35	24				
3	20	5	75	35	247	165		
4	30	6	139	48	584	270	2373	1392

Now assume that $m \geq 5$. Then, using the previous results, we know that $v_{2k+1}(g_0) > (m+1)m^k$ and $v_{2k}(g_1) < (m+1)^{k-1}(m^3 + 5m^2 + 8m + 4)/m^2$ for $0 < k$. We also utilize the well known inequality

$$\left(1 + \frac{1}{m}\right)^m < e \quad (2)$$

for all natural m [Finney and Thomas, 1994], where e is the basis of the natural logarithm. Then, it holds for $0 < k \leq m$ that

$$\begin{aligned} v_{2k}(g_1) &< (m+1)^{k-1}(m^3 + 5m^2 + 8m + 4)/m^2 \\ &= m^k(m+1)(1+1/m)^k(m^3 + 5m^2 + 8m + 4)/(m^2+m)^2 \\ &\leq m^k(m+1)(1+1/m)^m(m^3 + 5m^2 + 8m + 4)/(m^2+m)^2 \\ &< m^k(m+1)e(m^3 + 5m^2 + 8m + 4)/(m^2+m)^2 \\ &< m^k(m+1) \\ &< v_{2k+1}(g_0). \end{aligned}$$

2: $s = r_i$ for $0 < i \leq k \leq m$.

Value $x_{2k+1}(r_i)$: According to the induction hypothesis, it holds that $x_{2k+1}(r_i) = x_{2k}(r_i) + 1 = 2k - 2i + 3$.

Value $v_{2k+1}(r_i)$: According to the induction hypothesis, it holds that $v_{2k+1}(g_{i-1}) > v_{2k}(g_i)$ for $0 < i \leq k$. Thus, the next subroot that Node Counting visits is g_i. According to the lemma and the induction hypothesis, it holds that $v_{2k+1}(r_i) = \min(v_{2k}(g_i), v_{2k+1}(g_{i-1})) + x_{2k+1}(r_i) = v_{2k}(g_i) + 2k - 2i + 3$.

Inequality $v_{2k+1}(r_i) \geq v_{2k}(r_{i+1})$: For $i = k < m$, according to the induction hypothesis, it holds that $v_{2k+1}(r_i) = v_{2k}(g_i) + 2k - 2i + 3 \geq 1 = v_{2k}(r_{i+1})$ because only pass 0 reached subroot r_{i+1}. Otherwise, if $0 < i < k$ then it holds from the induction hypothesis that $v_{2k+1}(r_i) = v_{2k}(g_i) + 2k - 2i + 3 \geq v_{2k-1}(g_i) + 2k - 2i = v_{2k}(r_{i+1})$ because $v_p(s)$ is a non-decreasing function of pass number p.

3: $s = g_i$ for $0 < i < k \leq m$

Value $x_{2k+1}(g_i)$: According to the induction hypothesis, it holds that $x_{2k+1}(g_i) = x_{2k}(g_i) + 1 = 2k - 2i + 2$.

Value $v_{2k+1}(g_i)$: The induction hypothesis implies that $v_{2k}(r_{i+1}) \leq v_{2k+1}(r_i)$ so the next visited subroot is r_{i+1} and the lemma implies that $v_{2k+1}(g_i) = (m + i)v_{2k}(r_{i+1}) + x_{2k+1}(g_i) = (m + i)(v_{2k-1}(g_i) + 2k - 2i) + 2k - 2i + 2$.

Inequality $v_{2k+1}(g_i) > v_{2k}(g_{i+1})$ for $0 < i < k$: We start by noticing that $v_{2k+1}(g_i) = v_{2k}(g_i) + 1$ for $0 < i < k$. This can be shown be induction. For $0 < i = l < k$, it holds that $v_{2l+1}(g_i) = m + i + 2 = v_{2l}(g_i) + 1$. For $0 < i < l \leq k$, it holds that $v_{2l+1}(g_i) = (m+i)(v_{2l-1}(g_i)+2l-2i)+2l-2i+2 = (m + i)(v_{2l-2}(g_i) + 2l - 2i + 1) + 2l - 2i + 1 + 1 = v_{2l}(g_i) + 1$.

As a result, the postulated inequality $v_{2k+1}(g_i) > v_{2k}(g_{i+1})$ holds if and only if

$$v_{2k}(g_i) \quad \geq \quad v_{2k}(g_{i+1}). \tag{3}$$

According to the induction hypothesis, it holds for $0 < i \leq l \leq k$ that

$$v_{2l}(g_i) \quad = \quad \begin{cases} m + i + 1 & \text{for } i = l \\ (m + i)(v_{2l-2}(g_i) + 2l - 2i + 1) + 2l - 2i + 1 & \text{otherwise.} \end{cases}$$

From this definition, it follows that $v_{2l}(g_i) > (m + i)v_{2l-2}(g_i)$ for $0 < i < l \leq k$ and by induction on l we get $v_{2l}(g_i) > v_{2i}(g_i)(m+i)^{l-i} = (m+i+1)(m+i)^{l-i}$ for $0 < i < l \leq k$. Solving the recursion yields

$$\begin{aligned} v_{2k}(g_i) \quad &= \quad (m + i + 1) \\ &\quad \left((m + i)^{k-i} + \left((m + i)^{k-i} - 1\right) \frac{3(m + i) - 1}{(m + i - 1)^2} - (k - i)\frac{2}{m + i - 1} \right) \\ &< \quad (m + i + 1)(m + i)^{k-i} \left(1 + \frac{3}{m + i - 1} + \frac{2}{(m + i - 1)^2} \right). \end{aligned}$$

In the proof of this inequality we use again inequality (2), so for $m \geq 5,\ i > 0$ we obtain

$$
\begin{aligned}
v_{2k}(g_{i+1}) &< (m+i+2)(m+i+1)^{k-i-1}\left(1+\frac{3}{m+i}+\frac{2}{(m+i)^2}\right)\\
&= ((m+i+1)+1)(m+i+1)^{k-i-1}\left(1+\frac{3}{m+i}+\frac{2}{(m+i)^2}\right)\\
&= \left((m+i+1)^{k-i}+(m+i+1)^{k-i-1}\right)\left(1+\frac{3}{m+i}+\frac{2}{(m+i)^2}\right)\\
&= (m+i+1)^{k-i}\left(1+\frac{1}{m+i+1}\right)\left(1+\frac{3}{m+i}+\frac{2}{(m+i)^2}\right)\\
&= ((m+i)(1+\frac{1}{m+i}))^{k-i}\left(1+\frac{1}{m+i+1}\right)\left(1+\frac{3}{m+i}+\frac{2}{(m+i)^2}\right)\\
&= (m+i)^{k-i}(1+\frac{1}{m+i})^{k-i}\left(1+\frac{1}{m+i+1}\right)\left(1+\frac{3}{m+i}+\frac{2}{(m+i)^2}\right)\\
&\leq (m+i)^{k-i}(1+\frac{1}{m+i})^{m+i}\left(1+\frac{1}{5+1+1}\right)\left(1+\frac{3}{5+1}+\frac{2}{(5+1)^2}\right)\\
&< (m+i)^{k-i}e\frac{16}{9} \leq (m+i+1)(m+i)^{k-i}\frac{16e}{63}\\
&< (m+i+1)(m+i)^{k-i} < v_{2k}(g_i).
\end{aligned}
$$

proving the postulated inequality for $m \geq 5$. In the table below, we verify that $v_{2k}(g_i) \geq v_{2k}(g_{i+1})$ for $2 \leq m < 5$ using the solution of the recursion.

m	k	$v_{2k}(g_1)$	$v_{2k}(g_2)$	$v_{2k}(g_3)$	$v_{2k}(g_4)$
2	2	24	5		
3	2	35	6		
3	3	165	48	7	
4	2	48	7		
4	3	270	63	8	
4	4	1392	413	80	9

4: $s = g_k$ for $0 < k \leq m$

Value $x_{2k+1}(g_k)$: According to the induction hypothesis, it holds that $x_{2k+1}(g_k) = x_{2k}(g_k) + 1 = 2$.

Value $v_{2k+1}(g_k)$: Consider two cases. For $0 < k < m$, according to the induction hypothesis, it holds that $v_{2k+1}(r_k) \geq v_{2k}(r_{k+1})$ for $0 < k < m$. Thus, the next subroot that Node Counting visits is r_{k+1}. According to the lemma and the induction hypothesis, it holds that $v_{2k+1}(g_k) = (m+k)\min(v_{2k}(r_{k+1}), v_{2k+1}(r_k)) + x_{2k+1}(g_k) = m + k + 2$. The complementary case is for $k = m$, where, according the induction hypothesis, it also holds that $v_{2m+1}(r_m) = v_{2m}(g_m) + 3 \geq 1$. The value of the leaves of g_m is at least one, their value after the initial down pass. According to our selection rules, Node Counting moves directly to the goal state (and terminates) since the value of the start state is still one and the value of the goal state is still zero. According to the induction hypothesis, it holds that $v_{2m+1}(g_m) = v_{2m}(g_m) + 1 = 2m + 2$.

5: $s = r_{k+1}$ for $0 \leq k < m$

Value $x_{2k+1}(r_{k+1})$: According to the induction hypothesis, it holds that $x_{2k+1}(r_{k+1}) = x_{2k}(r_{k+1}) + 1 = 2$.

Value $v_{2k+1}(r_{k+1})$: It follows from the induction hypothesis, that $v_{2k}(g_{k+1}) = m + k + 2 \geq v_{2k+1}(g_k)$ because $v_{2k+1}(g_k) = m + k + 2$ for $0 < k < m$ and $v_{2k+1}(g_0) = v_{2k}(g_0) = m + 1$ for $k = 0$. According to our selection rules, this ends the up pass and starts a down pass. Thus, the next subroot that Node Counting visits is g_k. According to the lemma and the induction hypothesis, it holds that $v_{2k+1}(r_{k+1}) = \min(v_{2k}(g_{k+1}), v_{2k+1}(g_k)) + x_{2k+1}(r_{k+1}) = v_{2k+1}(g_k) + 2$.

6: $s = r_i$ for $1 \leq k + 1 < i \leq m$ or $s = g_i$ for $0 \leq k < i \leq m$.

Values: Since up pass $2k + 1$ starts at subroot r_0 and ends at subroot r_{k+1} (with the exception of up pass $2m + 1$ that ends at the goal state), Node Counting does not visit the subroots r_i for $i > k + 1$ nor the subroots g_i for $i > k$ during up pass $2k + 1$. Thus, according to the induction hypothesis, it holds that $x_{2k+1}(r_i) = x_{2k}(r_i) = 1$ and $v_{2k+1}(r_i) = v_{2k}(r_i) = 1$ for $i > k + 1$, and $x_{2k+1}(g_i) = x_{2k}(g_i) = 1$ and $v_{2k+1}(g_i) = v_{2k}(g_i) = m + i + 1$ for $i > k$.

Part 3: Assume that $p = 2k$ for $0 < k \leq m$. Down pass $2k$ starts where the previous up pass ended, that is, at subroot r_k. We distinguish the following five cases:

1: $s = r_k$ for $0 < k \leq m$.

Value $x_{2k}(r_k)$: According to the induction hypothesis, all previous passes, except passes 0 and $2k - 1$, ended before reaching r_k, so $x_{2k}(r_k) = x_{2k-1}(r_k) = 2$.

Value $v_{2k}(r_k)$: According to the induction hypothesis, Node Counting starts the down pass at subroot r_k and the next subroot that Node Counting visits is g_{k-1}. Thus, it holds that $v_{2k}(r_k) = v_{2k-1}(r_k) = v_{2k-1}(g_{k-1}) + 2$.

Inequality $v_{2k}(r_i) \geq v_{2k-1}(r_{i-1})$ for $1 < i = k$: As shown above, $v_{2k}(r_k) = v_{2k-1}(g_{k-1}) + 2$ and, according to the induction hypothesis, $v_{2k-1}(g_{k-1}) + 2 = m + k + 3 = v_{2k-2}(g_{k-1}) + 3 = v_{2k-1}(r_{k-1})$.

2: $s = g_i$ for $0 < i < k \leq m$.

Value $x_{2k}(g_i)$: According to the induction hypothesis, it holds that $x_{2k}(g_i) = x_{2k-1}(g_i) + 1 = 2k - 2i + 1$.

Value $v_{2k}(g_i)$: According to the induction hypothesis, it holds that $v_{2k}(r_{i+1}) \geq v_{2k-1}(r_i)$. Thus, according to our selection rules, Node Counting continues with the down pass and the next subroot that it visits is r_i. Thus, according to the lemma and the induction hypothesis, it holds that $v_{2k}(g_i) = (m + i)\min(v_{2k-1}(r_i), v_{2k}(r_{i+1})) + x_{2k}(g_i) = (m + i)v_{2k-1}(r_i) + 2k - 2i + 1 = (m + i)(v_{2k-2}(g_i) + 2k - 2i + 1) + 2k - 2i + 1$.

Inequality $v_{2k}(g_i) > v_{2k-1}(g_{i-1})$: According to the induction hypothesis, it holds that $v_{2k-2}(g_i) > v_{2k-3}(g_{i-1})$ for $0 < i < k - 1$, $v_{2k-2}(g_i) = v_{2k-2}(g_{k-1}) = m + k = v_{2k-3}(g_{k-2}) = v_{2k-3}(g_{i-1})$ for $1 < i = k - 1$, and $v_{2k-2}(g_i) = v_2(g_1) = m + 2 > m + 1 = v_0(g_0) = v_1(g_0) = v_{2k-3}(g_{i-1})$ for $1 = i = k - 1$. Thus, according to the induction hypothesis, $v_{2k}(g_i) = (m+i)(v_{2k-2}(g_i)+2k-$

$2i+1)+2k-2i+1 > (m+i-1)(v_{2k-3}(g_{i-1})+2k-2i)+2k-2i+2 = v_{2k-1}(g_{i-1})$ for $1 < i < k$. According to the induction hypothesis, it also holds that $v_{2k}(g_i) = v_{2k}(g_1) = (m+1)(v_{2k-2}(g_1)+2k-1)+2k-1 > m(v_{2k-3}(g_0)+2k-2)+k = m(v_{2k-4}(g_0)+2k-2)+k = v_{2k-2}(g_0) = v_{2k-1}(g_0) = v_{2k-1}(g_{i-1})$ for $1 = i < k$.

3: $s = r_i$ for $0 < i < k \leq m$.

Value $x_{2k}(r_i)$: According to the induction hypothesis, it holds that $x_{2k}(r_i) = x_{2k-1}(r_i)+1 = 2k-2i+2$.

Value $v_{2k}(r_i)$: According to the induction hypothesis, it holds that $v_{2k}(g_i) > v_{2k-1}(g_{i-1})$. Thus, the next subroot that Node Counting visits is g_{i-1}. According to the lemma and the induction hypothesis, it holds that $v_{2k}(r_i) = \min(v_{2k-1}(g_{i-1}), v_{2k}(g_i)) + x_{2k}(r_i) = v_{2k-1}(g_{i-1})+2k-2i+2$.

Inequality $v_{2k}(r_i) \geq v_{2k-1}(r_{i-1})$ for $1 < i < k$: According to the induction hypothesis, Node Counting visits subroot g_{i-1} during up pass $2k-1$. Thus, it holds that $v_{2k-1}(g_{i-1}) > v_{2k-2}(g_{i-1})$ and, according to the induction hypothesis, $v_{2k}(r_i) = v_{2k-1}(g_{i-1})+2k-2i+2 \geq v_{2k-2}(g_{i-1})+2k-2i+3 = v_{2k-1}(r_{i-1})$.

4: $s = g_0$ for $0 < k \leq m$.

Value $x_{2k}(g_0)$: According to the induction hypothesis, it holds that $x_{2k}(g_0) = x_{2k-1}(g_0)+1 = k+1$.

Value $v_{2k}(g_0)$: The next subroot that Node Counting visits is r_1, which ends the down pass and starts an up pass. For $k \geq 1$, according to the induction hypothesis, it holds that $v_{2k}(r_1) = v_{2k-1}(g_0)+2k = v_{2k-2}(g_0)+2k$. Thus, according to the lemma and the induction hypothesis, it holds that $v_{2k}(g_0) = m\, v_{2k}(r_1) + x_{2k}(g_0) = m\,(v_{2k-2}(g_0)+2k)+k+1$.

5: $s = r_i$ for $0 < k < i \leq m$ or $s = g_i$ for $0 < k \leq i \leq m$.

Values: Since down pass $2k$ starts at subroot r_k and ends at subroot g_0, Node Counting does not visit the subroots r_i for $i > k$ nor the subroots g_i for $i \geq k$ during down pass $2k$. Thus, according to the induction hypothesis, it holds that $x_{2k}(r_i) = x_{2k-1}(r_i) = 1$ and $v_{2k}(r_i) = v_{2k-1}(r_i) = 1$ for $i > k$, and $x_{2k}(g_i) = x_{2k-1}(g_i) = 1$ and $v_{2k}(g_i) = v_{2k-1}(g_i) = m+i+1$ for $i \geq k$.

This completes the proof. ■

Thus, Node Counting reaches the goal state during up pass $2m+1$. Setting $l = m$ in Equation (1) results in

$$v_{2m+1}(g_0) = v_{2m}(g_0) \quad = \quad \frac{m^{m+3}+m^{m+2}+m^{m+1}-2m^3-m^2-m+1}{m^2-2m+1} > m^m.$$

For example, $v_5(g_0) = 35$ for $m = 2$ as shown in Figure 2.

Consider an arbitrary constant $0 < \epsilon < 1/6$ and assume that $m > \max\left(\frac{1}{\epsilon} - 4, \left(\frac{3}{2-8\epsilon}\right)^{1/\epsilon}\right) \geq 2$. Note that $n \geq m$ for our trees. Then,

$$
\begin{aligned}
n &= \tfrac{3}{2}m^2 + \tfrac{9}{2}m + 3 \\
&< \tfrac{3}{2}\left(1 + \tfrac{4}{m}\right) m^2 && \text{since } m > 2 \\
&< \tfrac{3}{2}\left(1 + \tfrac{4\epsilon}{1-4\epsilon}\right) m^2 && \text{since } m > \tfrac{1}{\epsilon} - 4 \\
&= \tfrac{3}{2}\tfrac{m^2}{1-4\epsilon}
\end{aligned}
$$

and thus $m > \sqrt{\frac{2}{3}n(1-4\epsilon)} > 1$ (A). In the following, we also utilize that $(an)^k > n^{(1-\epsilon)k}$ for $n > \left(\frac{1}{a}\right)^{1/\epsilon}$ and arbitrary constants $a > 0$ and $k > 0$ (B). Then,

$$
\begin{aligned}
m^m &> \left(\sqrt{\tfrac{2}{3}n(1-4\epsilon)}\right)^{\sqrt{\frac{2}{3}n(1-4\epsilon)}} && \text{since (A)} \\
&= \left(\left(\tfrac{2}{3} - \tfrac{8}{3}\epsilon\right) n\right)^{\frac{1}{2}\sqrt{\frac{2}{3}n(1-4\epsilon)}} \\
&> n^{(1-\epsilon)\frac{1}{2}\sqrt{\frac{2}{3}n(1-4\epsilon)}} && \text{since (B)} \\
&= n^{\sqrt{(1-\epsilon)^2\frac{1}{6}n(1-4\epsilon)}} \\
&> n^{\sqrt{(1-2\epsilon)\frac{1}{6}n(1-4\epsilon)}} \\
&= n^{\sqrt{\left(\frac{1}{6}-\epsilon+\frac{4}{3}\epsilon^2\right)n}} \\
&> n^{\sqrt{\left(\frac{1}{6}-\epsilon\right)n}}
\end{aligned}
$$

Using the two inequalities above, it holds that $v_{2m+1}(g_0) > m^m > n^{\sqrt{\left(\frac{1}{6}-\epsilon\right)n}}$ for $m > \max\left(\frac{1}{\epsilon} - 4, \left(\frac{3}{2-8\epsilon}\right)^{1/\epsilon}\right)$ and thus also for sufficiently large n. $v_{2m+1}(g_0)$ is the value of g_0 after Node Counting terminates on a tree with $m+1$ levels. This value equals the number of times Node Counting has visited g_0, which is a lower bound on the number of actions it has executed. Thus, the complexity of Node Counting on undirected graphs is $\Omega\left(n^{\sqrt{(1/6-\epsilon)n}}\right)$.

References

(Balch and Arkin, 1993) Balch, T. and Arkin, R. 1993. Avoiding the past: A simple, but effective strategy for reactive navigation. In *International Conference on Robotics and Automation*. 678–685.

(Finney and Thomas, 1994) Finney, R. and Thomas, G. 1994. *Calculus*. Addison Wesley.

(Thrun, 1992) Thrun, S. 1992. The role of exploration in learning control with neural networks. In White, D. and Sofge, D., editors 1992, *Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches*. Van Nostrand Reinhold. 527–559.

(Wagner *et al.*, 1997) Wagner, I.; Lindenbaum, M.; and Bruckstein, A. 1997. On-line graph searching by a smell-oriented vertex process. In *Proceedings of the AAAI Workshop on On-Line Search*. 122–125.