Proofs of bounds on the “hop” algorithm for searching worms.

Note: Two pages prove # hops for a worm of length \( l \) is \( \Omega(\sqrt{l}) \) and \( O(l^{2/3}) \).

1. Consider a worm of length \( l \).

2. Note than in an infinite grid, the number of squares at distance \( d > 0 \) from a given square is \( 4d \).

3. Consider a traversal of a worm made by the “hop” algorithm in searching for a square not actually in the worm. If \( n \) is the longest hop in a traversal and \( H \) the number of hops, we have that

\[
H \leq \sum_{d=1}^{n} 4d = 4 \left( \frac{n(n+1)}{2} \right) = 2n^2 + 2n,
\]

because you can’t have more hops of length \( d \) than there are squares at distance \( d \) from the target \((x, y)\), and there are exactly \( 4d \) squares at distance \( d \) from \((x, y)\). Similarly, the worm body length satisfies

\[
l \geq \sum_{d=1}^{n} 4d \cdot d = 4 \sum_{d=1}^{n} d^2 = \Theta(n^3), \quad \text{i.e. } l = \Omega(n^3).
\]

4. **Theorem:** In an \( H \)-hop traversal, the longest hop, \( n \), satisfies

\[
n \geq \left\lceil -1 + \sqrt{1 + 2H} \right\rceil.
\]

**Proof:** From the above, we know that \( H \leq 2n^2 + 2 \), which we can rewrite as \( n^2 + n - H/2 \geq 0 \). Of course we’re only interested in positive values of \( n \), and by the quadratic formula the only positive root of \( n^2 + n - H/2 \) is \( -1 + \sqrt{1 + 2H} \) and therefore \( n \geq -1 + \sqrt{1 + 2H} \). Finally, since \( n \) is an integer, we get

\[
n \geq \left\lceil -1 + \sqrt{1 + 2H} \right\rceil.
\]

5. Combining the two previous results, \( l = \Omega(n^3) \) and \( \left\lceil -1 + \sqrt{1 + 2H}/2 \right\rceil \)
we get

\[
l = \Omega \left( \left(-1 + \sqrt{1 + 2H}/2 \right)^3 \right) = \Omega \left( H^{3/2} \right).
\]

6. Thus, by the def. of \( \Omega \), for some constant \( a > 0 \), when \( H \) is large we have

\[
l \geq aH^{3/2} \quad \text{which means } (1/a)^{2/3} l^{2/3} \geq H \quad \text{which means } H = O\left(l^{2/3}\right).
\]

---

\(^1\)See page 1060 of the textbook.
Consider a worm that is wrapped around cell \((x, y)\), so that it spirals away. We should describe this a bit precisely. Let a \textit{d-square} be the square of cells \((i, j)\) formed by the rows \(j = x + d\) and \(j = x - d\), and the columns \(i = x + d\) and \(i = x - d\). The worm starts at cell \((x, y - 1)\) and moves counter-clockwise around the 1-square until it uses up all the cells in the 1-square. Then it crosses into the 2-square and moves counter-clockwise around the 2-square until it uses up all those squares. We’ll call this a “spiral worm”.

**Theorem:** The “hop” algorithm makes \(\Theta(\sqrt{n})\) hops in traversing the spiral worm around the point \((x, y)\) it is searching for.

**Proof:** First note that a \textit{d}-square consists of \(8d\) cells. This may take a bit of thinking to convince yourself of. Next note the distance of any cell in a \textit{d}-square from \((x, y)\) is between \(d\) and \(2d\).

Consider your last hop from a cell in a \textit{d}-square. The hop length is at most \(2d\). Since that is less than the \(8d\) cells that make up the \((d + 1)\)-square you move into, you end your hop in the \((d + 1)\)-square (as opposed to moving all the way around and out of it during that single hop). So, if you land in a \textit{d}-square, you will eventually land in a \((d + 1)\)-square. Since our first hop lands in a 1-square, induction tells us that we land in every \textit{d}-square the worm fills up.

So we have at least one landing in every \textit{d}-square filled up by the worm, and clearly no more than 8, since each jump has length at least \(d\) and there are only \(8d\) cells in the \textit{d}-square. If the worm fills up the \textit{d}-squares from \(d = 1..k\). Then the number of hops, \(H\), satisfies

\[
k \leq H \leq 8k + 7, \quad ^{2}
\]

i.e. \(H = \Theta(k)\). Meanwhile, the worm body has length \(l\), where

\[
l = \sum_{d=1}^{k} 8d + \text{spillover} = 4k(k + 1) + \text{spillover}
\]

which means \(k = \Theta(\sqrt{l})\). We have \(H = \Theta(k)\) and \(k = \Theta(\sqrt{l})\), so \(H = \Theta(\sqrt{l})\).

Since we proved the number of hops is \(\Theta(\sqrt{l})\) for a particular worm configuration, i.e. the spiral, the worst case can’t be better and we get that the “hop” algorithm is \(\Omega(\sqrt{l}n)\).

\(^{2}\)The +7 is because the worm might fill some, though not all, of the \((k + 1)\)-square.