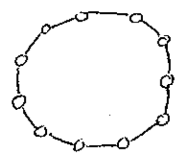
Def: Given (X_t), Ω, the <u>cover time</u> τ_{cov} is the first time that all states have been visited.

We are intersted in the expected value of τ_{cov} from the "worst case" Starting position. Formally, this is

$$t_{cov} = \max_{x \in \Omega} E_x \tau_{cov}$$

eg 1: Random walk on n-cycle

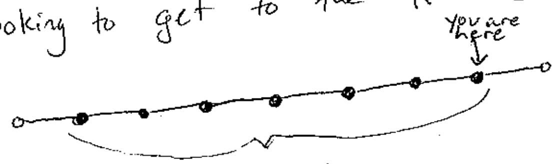


This can be thought of as a random walk on $\mathbb{Z}$ mod n.

t_{cov} is the same as the expected time it takes to visit n states in a walk on $\mathbb{Z}$, since they will all be contiguous and therefore different mod n.

How long does it take to visit a new state?

If we have just reached the $(K-1)^{th}$ state for the first time, and are looking to get to the K^{th} state, we have this picture.



we are at the end of the contiguous region and want to get to one of the white states. This is the gamblers ruin problem where we start with K-1 dollars and stop at K or when we are broke. The expected time is $(1)(K-1) = K-1$.

so the expected time to visit n new states is $\sum_{K=1}^{n} K-1 = \frac{n(n-1)}{2}$

(7 min

General Bound on the cover time (Matthews Method)

Recall: If we look at any two points $a, b \in \Omega$ we can ask the expected time to go from a to b. The max over all pairs is t_{hit}. Formally,

$$t_{hit} := \max_{a,b \in \Omega} E_a \tau_b$$

The bound for the cover time is similar to the coupon collector problem, except instead of taking expected time n to collect a particular coupon it takes $\leq t_{hit}$, and instead of taking $n \log n$ to collect all of them, it takes $t_{hit} \log n$.

Pf: Assume $\Omega = [n]$ and let $x \in \Omega$ be an arbitrary starting state, and choose $\sigma \in S_n$ uniformly at random. (Think of σ as an ordering of the states, we will look to collect the states in σ order).

Let T_k be the first time that we have collected all of $\sigma(1), \sigma(2), \ldots, \sigma(k)$. (So the quanitity we are looking for is $t_{cov} = E_x[T_n]$.)

What is $E[T_1]$?

$$E_x[T_1] = E_x \tau_{\sigma(1)} \leq t_{hit}.$$

~~[illegible]~~

~~[illegible]~~

~~[illegible]~~

Do to this more formally, we need excersize 11.1a

<u>Claim:</u> Let Y be a random variable on a probability space and let $B = \bigcup_j B_j$ be a partition of B.

Then if $E(Y \mid B_j) \leq M$ for all j, then $E(Y \mid B) \leq M$

<u>Pf:</u>

$$E(Y \mid B) = \frac{1}{P(B)} \sum P(B_j)\, E(Y \mid B_j) \leq \frac{1}{P(B)} \sum P(B_j) M = M$$

(average the values of Y, weighted by $P(B_j)$)

So if $s \in \Omega$,

$$E_x(T_1 \mid \sigma(1) = s) = E_x \tau_s \leq t_{hit}.$$

and since the events $\sigma(1) = s$ partition the space S_n,

$$E_x(T_1) = E_x(T_1 \mid \sigma \in S_n) \leq t_{hit}.$$

~~Now we~~

What is $E_x[T_2 - T_1]$?

If we hit $\sigma(2)$ before $\sigma(1)$, then $T_2 - T_1 = 0$.

Else, by similar reasoning as before, $E[T_2 - T_1] = E_{\sigma(1)} \tau_{\sigma(2)} \leq t_{hit}$.

Let $r, s \in \Omega$, and WLOG, assume our chain hits r before s.

Then

$$E\left(T_2 - T_1 \mid \{\sigma(1), \sigma(2)\} = \{r, s\}\right) = \frac{1}{2} E\left[T_2 - T_1 \,\middle|\, \begin{matrix}\sigma(1) = r \\ \sigma(2) = s\end{matrix}\right] + \frac{1}{2} E\left[T_2 - T_1 \,\middle|\, \begin{matrix}\sigma(1) = s \\ \sigma(2) = r\end{matrix}\right]$$

$$\leq \frac{1}{2} \cdot \tau_{hit} + \frac{1}{2} \cdot 0$$

(since we chose σ uniformly at random)

Again, by excercise 11.1a, this shows that $E[T_2 - T_1] \leq \frac{1}{2} \tau_{hit}$

So $E[T_2] = E[T_1] + E[T_2 - T_1] \leq \tau_{hit}\left(1 + \frac{1}{2}\right)$

What is $E[T_3 - T_2]$?

If we hit $\sigma(3)$ before $\sigma(1)$ or $\sigma(2)$ then $T_3 - T_2 = 0$.

Else, $E[T_3 - T_2] = E_{\sigma(2)} \tau_{\sigma(3)} \leq t_{hit}$

Let $q, r, s \in \Omega$ and WLOG assume we hit q then r then s.

$$E\left(T_3 - T_2 \mid \{\sigma(1), \sigma(2), \sigma(3)\} = \{q, r, s\}\right)$$

$$= \frac{1}{3} \cdot E\left(T_3 - T_2 \;\middle|\; \begin{matrix}\sigma(3) = s \\ \{\sigma(1), \sigma(2)\} = \{q, r\}\end{matrix}\right) + \frac{2}{3} E\left(T_3 - T_2 \;\middle|\; \begin{matrix}\sigma(3) \in \{q, r\} \\ \{\sigma(1), \sigma(2), \sigma(3)\} = \{q, r, s\}\end{matrix}\right)$$

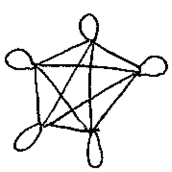
$$= \frac{1}{3} \cdot t_{hit} \quad + \frac{2}{3} \cdot 0$$

So $E[T_3] = E[T_1] + E[T_2 - T_1] + E[T_3 - T_2]$

$$= t_{hit}\left(1 + \frac{1}{2} + \frac{1}{3}\right)$$

by induction, we can show that $E[T_n] = t_{hit}\left(1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n}\right)$

Eg2: Random walk on complete graph with self loops



for all states $x, y \in \Omega$, $E_x \tau_y = t_{hit}$,
so in the previous proof we can replace all the $\leq$ with $=$, so our bound is tight.

Lower Bounds

Let $A \subseteq \Omega$ and $t_{min}^{A} = \min_{a,b \in A} E_a \tau_b$.

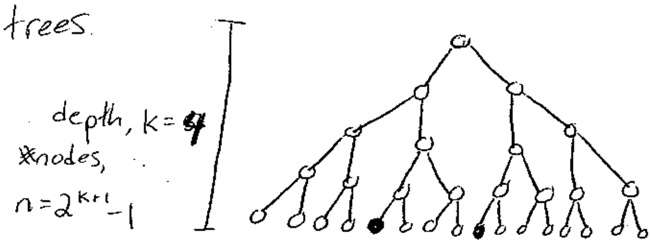
then $t_{cov} \geq t_{cov}^{A}$ (it takes at least as long to cover Ω as it does to cover A)

and by a similar coupon collecting argument, we can show

$$t_{cov} \geq t_{cov}^{A} \geq t_{min}^{A} \left(1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{|A|-1}\right)$$

which gives us a lower bound for each set $A \subseteq \Omega$. There is some cleverness required for choosing A because if you make it too small then $\left(1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{|A|-1}\right)$ will be small, but if you make A big, then you will be able to minimize t_{min}^{A} more.

<u>eg3</u>: Random walk on binary trees



which nodes maximize t_{hit}?

2 nodes who's only common ancestor is the root.
The expected time to get between these two nodes is the same as the time to get from a leaf, to the root, and back to that same leaf. This is the commute time from the root to the leaf.

<u>Recall</u>: The commute time, the time it takes to get from state a to state b and back to state a is

$$E_a(\tau_{a,b}) = C_G \cdot R(a \leftrightarrow b) \qquad \text{(prop 10.6)}$$

So

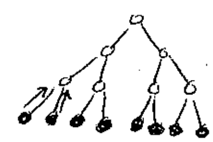
$$E_{Leaf}(\tau_{leaf,\,root}) = 2(\#\text{edges}) \cdot (\text{depth}) = 2(n-1)\cdot k$$

$$\text{so } t_{cov} \leq 2(n-1)\cdot k\left(1+\tfrac{1}{2}+\dots+\tfrac{1}{n}\right) = (2+o(1))\,(n-1)\,k \cdot \log 2^{k+1}$$

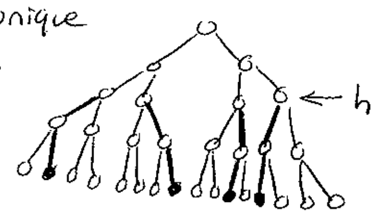
$$= (2+o(1))\cdot(n)\cdot k^2 \cdot \log 2$$

$$= (2+o(1))\cdot \log 2 \cdot n k^2.$$

To get a lower bound, we need to choose a good A. We dont want A = {leaves} because then we will get a small t^A_{min} by choosing two leaves with the same parent.



Instead let h be some height, and let A be a set of 2^h leaves, each with a unique ancestor at height h. We will decide how to choose h later.



The closest two vertices have a common ancestor at height h-1, so the commute time identity gives

$$t^A_{min} = 2(n-1)\cdot(k-(h-1))$$

again, $t^{A}_{min} = 2(n-1)\left(k-(h+1)\right)$, and $|A| = 2^h$

so $t_{cov} \geqslant 2\cdot(n-1)\cdot(k-h+1)\cdot\left(1+\frac{1}{2}+\frac{1}{3}+\dots+\frac{1}{|A|-1}\right)$

$= (2+o(1))\cdot n(k-h)\cdot h\cdot\log 2$

which is maximized when $h=\frac{k}{2}$. then

$t_{cov} \geqslant \frac{1}{4}(2+o(1))(\log 2)nk^2$

(our upper bound was the same, without the $\frac{1}{4}$)

eg 4: Random walk on d dimensional torus

Case 1: $d \geq 3$

Recall: (prop 10.13) There exist constants c_d, C_d such that

$$c_d n^d \leq E_x(\tau_y) \leq C_d n^d$$

these bounds do not depend on x and y

Thus $t_{cov} \leq C_d n^d \cdot \left(1 + \frac{1}{2} + \dots + \frac{1}{n^d}\right) = C_d\, d\, (\log n) \cdot n^d \cdot (1 + o(1))$

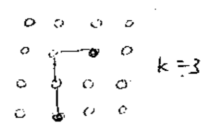
Taking $A = \mathbb{Z}_n^d$ (the entire Torus)

then $t_{cov} \geq c_d n^d \cdot \left(1 + \frac{1}{2} + \dots + \frac{1}{n^d - 1}\right) = c_d \cdot d \cdot (\log n) \cdot n^d (1 + o(1))$

Case 2: $d = 2$

Recall: (prop 10.13) If $x, y \in \mathbb{Z}_n^d$ are distance k apart, then

$$c_2 n^2 \cdot \log k \leq E_x \tau_y \leq C_2 n^2 \cdot \log k$$

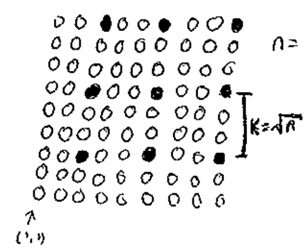


so since the max k can be is n

$$t_{cov} \leq C_2\, n^2 \cdot \log n \left(1 + \frac{1}{2} + \dots + \frac{1}{n^2}\right)$$
$$= (1 + o(1))\, C_2\, n^2 \cdot 2\, (\log n)^2$$

for lower bound, Assume n is a perfect square, and let $A = \{$Point with coordinates that are multiples of $\sqrt{n}\}$

so $|A| = n$, and $t_{min}^{A} = c_2 \cdot n^2 \cdot \log\sqrt{n}$



so $t_{cov} \geq c_2 \cdot n^2 \cdot \log\sqrt{n} \cdot \left(1 + \frac{1}{2} + \dots + \frac{1}{n-1}\right)$

$$= (1 + o(1))\, c_2 n^2 \cdot \frac{1}{2} \log n \cdot \log n$$
$$= (1 + o(1))\, c_2 \cdot \frac{1}{2} \cdot n^2 (\log n)^2$$