6.5 Probability Calculations in Hashing

We can use our knowledge of probability and expected values to analyze a number of interesting aspects of hashing including:

1. expected number of items per location,
2. expected time for a search,
3. expected number of collisions,
4. expected number of empty locations,
5. expected time until all locations have at least one item,
6. expected maximum number of items per location.

Expected Number of Items per Location

Exercise 6.5-1 We are going to compute the expected number of items that hash to any particular location in a hash table. Our model of hashing n items into a table of size k allows us to think of the process as n independent trials, each with k possible outcomes (the k locations in the table). On each trial we hash another key into the table. If we hash n items into a table with k locations, what is the probability that any one item hashes into location 1? Let X_i be the random variable that counts the number of items that hash to location 1 in trial i (so that X_i is either 0 or 1). What is the expected value of X_i? Let X be the random variable $X_1 + X_2 + \cdots + X_n$. What is the expected value of X? What is the expected number of items that hash to location 1? Was the fact that we were talking about location 1 special in any way? That is, does the same expected value apply to every location?

Exercise 6.5-2 Again we are hashing n items into k locations. Our model of hashing is that of Exercise 6.5-1. What is the probability that a location is empty? What is the expected number of empty locations? Suppose we now hash n items into the same number n of locations. What limit does the expected fraction of empty places approach as n gets large?

In Exercise 6.5-1, the probability that any one item hashes into location 1 is $1/k$, because all k locations are equally likely. The expected value of X_i is then $1/k$. The expected value of X is then n/k, the sum of n terms each equal to $1/k$. Of course the same expected value applies to any location. Thus we have proved the following theorem.

Theorem 6.13 *In hashing n items into a hash table of size k, the expected number of items that hash to any one location is n/k.*

Expected Number of Empty Locations

In Exercise 6.5-2 the probability that position i will be empty after we hash 1 item into the table will be $1 - \frac{1}{k}$. (Why?) In fact, we can think of our process as an independent trials process with two outcomes: the key hashes to slot i or it doesn't. From this point of view, it is clear that the probability of nothing hashing to slot i in n trials is $(1 - \frac{1}{k})^n$. Now consider the original sample space again and let X_i be 1 if slot i is empty for a given sequence of hashes or 0 if it is not. Then the number of empty slots for a given sequence of hashes is $X_1 + X_2 + \cdots + X_k$ evaluated at that sequence. Therefore, the expected number of empty slots is, by Theorem 6.9, $k(1 - \frac{1}{k})^n$. Thus we have proved another nice theorem about hashing.

Theorem 6.14 *In hashing n items into a hash table with k locations, the expected number of empty locations is $k(1 - \frac{1}{k})^n$.*

Proof: Given above.■

If we have the same number of slots as places, the expected number of empty slots is $n(1-\frac{1}{n})^n$, so the expected fraction of empty slots is $(1-\frac{1}{n})^n$. What does this fraction approach as n grows? You may recall that $\lim_{n\to\infty}(1+\frac{1}{n})^n$ is e, the base for the natural logarithm. In the problems at the end of the section, we show you how to derive from this that $\lim_{n\to\infty}(1-\frac{1}{n})^n$ is e^{-1}. Thus for a reasonably large hash table, if we hash in as many items as we have slots, we expect a fraction $1/e$ of those slots to remain empty. In other words, we expect n/e empty slots. On the other hand, we expect $\frac{n}{n}$ items per location, which suggests that we should expect each slot to have an item and therefore expect to have no empty locations. Is something wrong? No, but we simply have to accept that our expectations about expectation just don't always hold true. What went wrong in that apparent contradiction is that our definition of expected value doesn't imply that if we have an expectation of one key per location then every location must have a key, but only that empty locations have to be balanced out by locations with more than one key. When we want to make a statement about expected values, we must use either our definitions or theorems to back it up. This is another example of why we have to back up intuition about probability with careful analysis.

Expected Number of Collisions

We say that we have a *collision* when we hash an item to a location that already contains an item. How can we compute the expected number of collisions? The number of collisions will be the number n of keys hashed minus the number of occupied locations because each occupied location will contain one key that will not have collided in the process of being hashed. Thus, by Theorems 6.9 and 6.10,

$$E(\text{collisions}) = n - E(\text{occupied locations}) = n - k + E(\text{empty locations}) \tag{6.26}$$

where the last equality follows because the expected number of occupied locations is k minus the expected number of unoccupied locations. This gives us yet another theorem.

Theorem 6.15 *In hashing n items into a hash table with k locations, the expected number of collisions is $n - k + k(1 - \frac{1}{k})^n$.*

Proof: We have already shown in Theorem 2 that the expected number of empty locations is $k(1-\frac{1}{k})^n$. Substituting this into Equation 6.26 gives our formula. ■

Exercise 6.5-3 In real applications, it is often the case that the hash table size is not fixed in advance, since you don't know, in advance, how many items you will insert. The most common heuristic for dealing with this is to start k, the hash table size, at some reasonably small value, and then when n, the number of items gets to be greater than $2k$, you double the hash table size. In this exercise, we propose a different idea. Suppose you waited until every single slot in the hash table had at least one item in it, and then you increased the table size. What is the expected number of items that will be in the table when you increase the size? In other words, how many items do you expect to insert into a hash table in order to ensure that every slot has at least one item? (Hint: Let X_i be the number of items added between the time that there are $i-1$ occupied slots for the first time and the first time that there are i occupied slots.)

For Exercise 6.5-3, the key is to let X_i be the number of items added between the time that there are $i-1$ full slots for the first time and i full slots for the first time. Let's think about this random variable. $E(X_1) = 1$, since after one insertion there is one full slot. In fact X_1 itself is equal to 1.

To compute the expected value of X_2, we note that X_2 can take on any value greater than 1. In fact if we think about it, what we have here (until we actually hash an item to a new slot) is an independent trials process with two outcomes, with success meaning our item hashes to an unused slot. X_2 counts the number of trials until the first success. The probability of success is $(k-1)/k$. In asking for the expected value of X_2, we are asking for expected number of steps until the first success. Thus we can apply Lemma 6.12 to get that it is $k/(k-1)$.

Continuing, X_3 similarly counts the number of steps in an independent trials process (with two outcomes) that stops at the first success and has probability of success $(k-2)/k$. Thus the expected number of steps until the first success is $k/(k-2)$.

In general, we have that X_i counts the number of trials until success in an independent trials process with probability of success $(k-i+1)/k$ and thus the expected number of steps until the first success is $k/(k-i+1)$, which is the expected value of X_i.

The total time until all slots are full is just $X = X_1 + \cdots + X_k$. Taking expectations and using Lemma 6.12 we get

$$\begin{aligned} E(X) &= \sum_{j=1}^{k} E(X_j) \\ &= \sum_{j=1}^{k} \frac{k}{k-j+1} \\ &= k\sum_{j=1}^{k} \frac{1}{k-j+1} \\ &= k\sum k-j+1 = 1^k \frac{1}{k-j+1} \\ &= k\sum_{i=1}^{k} \frac{1}{i}, \end{aligned}$$

where the last line follows just by switching the variable of the summation, that is, letting $k - j + 1 = i$ and summing over i.[6] Now the quantity $\sum_{i=1}^{k} \frac{1}{i}$ is known as a *harmonic number*, and is sometimes denoted by H_k. It is well known (and you can see why in the problems at the end of the section) that $\sum_{i=1}^{k} \frac{1}{i} = \Theta(\log k)$, and more precisely

$$\frac{1}{4} + \ln k \leq H_k \leq 1 + \ln k, \tag{6.27}$$

and in fact,

$$\frac{1}{2} + \ln k \leq H_k \leq 1 + \ln k, \tag{6.28}$$

when k is large enough. As n gets large, $H_n - \ln n$ approaches a limit called *Euler's constant*; Euler's constant is about .58. Equation 6.27 gives us that $E(X) = O(k \log k)$.

Theorem 6.16 *The expected number of items needed to fill all slots of a hash table of size k is between $k \ln k + \frac{1}{2}k$ and $k \ln k + k$.*

Proof: Given above.■.

So in order to fill every slot in a hash table of size k, we need to hash roughly $k \ln k$ items. This problem is sometimes called the *coupon collectors problem.*

Expected maximum number of elements in a slot of a hash table

In a hash table, the time to find an item is related to the number of items in the slot where you are looking. Thus an interesting quantity is the expected maximum length of the list of items in a slot in a hash table. This quantity is more complicated than many of the others we have been computing, and hence we will only try to upper bound it, rather than compute it exactly. In doing so, we will introduce a few upper bounds and techniques that appear frequently and are useful in many areas of mathematics and computer science. We will be able to prove that if we hash n items into a hash table of size n, the expected length of the longest list is $O(\log n / \log \log n)$. One can also prove, although we won't do it here, that with high probability, there will be some list with $\Omega(\log n / \log \log n)$ items in it, so our bound is, up to constant factors, the best possible.

Before we start, we give some useful upper bounds. The first allows us to bound terms that look like $(1 + \frac{1}{x})^x$, for any positive x, by e.

Lemma 6.17 *For all $x > 0$, $(1 + \frac{1}{x})^x \leq e$.*

Proof: $\lim_{x \to \infty} (1 + \frac{1}{x})^x = e$, and $1 + (\frac{1}{x})^x$ has positive first derivative. ■

Second, we will use an approximation called Stirling's formula,

$$x! = \left(\frac{x}{e}\right)^x \sqrt{2\pi x}(1 + \Theta(1/n)),$$

[6] note that $k - j + 1$ runs from k to 1 as j runs from 1 to k, so we are describing exactly the same sum.

which tells us, roughly, that $(x/e)^x$ is a good approximation for $x!$. Moreover the constant in the $\Theta(1/n)$ term is extremely small, so for our purposes we will just say that

$$x! = \left(\frac{x}{e}\right)^x \sqrt{2\pi x}.$$

(We use this equality only in our proof of Lemma 6.18. You will see in that Lemma that we make the statement that $\sqrt{2\pi} > 1$. In fact, $\sqrt{2\pi} > 2$, and this is more than enough to make up for any lack of accuracy in our approximation.) Using Stirling's formula, we can get a bound on $\binom{n}{t}$,

Lemma 6.18 *For $n > t > 0$,*

$$\binom{n}{t} \leq \frac{n^n}{t^t(n-t)^{n-t}}.$$

Proof:

$$\begin{aligned}
\binom{n}{t} &= \frac{n!}{t!(n-t)!} && (6.29)\\
&= \frac{(n/e)^n\sqrt{2\pi n}}{(t/e)^t\sqrt{2\pi t}((n-t)/e)^{n-t}\sqrt{2\pi(n-t)}} && (6.30)\\
&= \frac{n^n\sqrt{n}}{t^t(n-t)^{n-t}\sqrt{2\pi}\sqrt{t(n-t)}} && (6.31)
\end{aligned}$$

Now if $1 < t < n-1$, we have $t(n-t) \geq n$, so that $\sqrt{t(n-t)} \geq \sqrt{n}$. Further $\sqrt{2\pi} > 1$. We can use these to facts to upper bound the quantity marked 6.31 by

$$\frac{n^n}{t^t(n-t)^{n-t}}$$

When $t = 1$ or $t = n-1$, the inequality in the statement of the lemma is $n \leq n^n/(n-1)^{n-1}$ which is true since $n - 1 < n$. ■

We are now ready to attack the problem at hand, the expected value of the maximum list size. Let's start with a related quantity that we already know how to compute. Let H_{it} be the event that t keys hash to slot i. $P(H_{it})$ is just the probability of t successes in an independent trials process with success probability $1/n$, so

$$P(H_{it}) = \binom{n}{t}\left(\frac{1}{n}\right)^t\left(1-\frac{1}{n}\right)^{n-t}. \quad (6.32)$$

Now we relate this known quantity to the probability of the event M_t that the maximum list size is t.

Lemma 6.19 *Let M_t be the event that t is the maximum list size in hashing n items into a hash table of size n. Let H_{1t} be the event that t keys hash to position 1. Then*

$$P(M_t) \leq nP(H_{1t})$$

Proof: We begin by letting M_{it} be the event that the maximum list size is t and this list appears in slot i. Observe that that since M_{it} is a subset of H_{it},

$$P(M_{it}) \leq P(H_{it}). \tag{6.33}$$

We know that, by definition,

$$M_t = M_{1t} \cup \cdots \cup M_{nt},$$

and so

$$P(M_t) = P(M_{1t} \cup \cdots \cup M_{nt}).$$

Therefore, since the sum of the probabilities of the individual events must be at least as large as the probability of the union,

$$P(M_t) \leq P(M_{1t}) + P(M_{2t}) + \cdots + P(M_{nt}). \tag{6.34}$$

(Recall that we introduced the Principle of Inclusion and Exclusion because the right hand side overestimated the probability of the union. Note that the inequality in Equation 6.34 holds for any union, not just this one: it is sometimes called *Boole's inequality.*)

In this case, for any i and j, $P(M_{it}) = P(M_{jt})$, since there is no reason for slot i to be more likely than slot j to be the maximum. We can therefore write that

$$P(M_t) = nP(M_{1t}) \leq nP(H_{1t}).$$

■

Now we can use Equation 6.32 for $P(H_{1t})$ and then apply Lemma 6.18 to get that

$$\begin{aligned} P(H_{1t}) &= \binom{n}{t}\left(\frac{1}{n}\right)^t\left(1-\frac{1}{n}\right)^{n-t} \\ &\leq \frac{n^n}{t^t(n-t)^{n-t}}\left(\frac{1}{n}\right)^t\left(1-\frac{1}{n}\right)^{n-t}. \end{aligned}$$

We continue, using algebra, the fact that $(1-\frac{1}{n})^{n-t} \leq 1$ and Lemma 6.17 to get

$$\begin{aligned} &\leq \frac{n^n}{t^t(n-t)^{n-t}n^t} \\ &= \frac{n^{n-t}}{t^t(n-t)^{n-t}} \\ &= \left(\frac{n}{n-t}\right)^{n-t}\frac{1}{t^t} \\ &= \left(1+\frac{t}{n-t}\right)^{n-t}\frac{1}{t^t} \\ &= \left(\left(1+\frac{t}{n-t}\right)^{\frac{n-t}{t}}\right)^t\frac{1}{t^t} \\ &\leq \frac{e^t}{t^t}. \end{aligned}$$

We have shown the following:

Lemma 6.20 *$P(M_t)$, the probability that the maximum list length is t, is at most ne^t/t^t.*

Proof: Our sequence of equations and inequalities above showed that $P(H_{1t}) \leq \frac{e^t}{t^t}$. Multiplying by n and applying Lemma 6.19 gives our result.■

Now that we have a bound on $P(M_t)$ we can compute a bound on the expected length of the longest list, namely

$$\sum_{t=0}^{n} P(M_t)t.$$

However, if we think carefully about the bound in Lemma 6.20, we see that we have a problem. For example when $t = 1$, the lemma tells us that $P(M_1) \leq ne$. This is vacuous, as we know that any probability is at most 1, We could make a stronger statement that $P(M_t) \leq \max\{ne^t/t^t, 1\}$, but even this wouldn't be sufficient, since it would tell us things like $P(M_1) + P(M_2) \leq 2$, which is also vacuous. All is not lost however. Our lemma causes this problem only when t is small. We will split the sum defining the expected value into two parts and bound the expectation for each part separately. The intuition is that when we restrict t to be small, then $\sum P(M_t)t$ is small because t is small (and over all t, $\sum P(M_t) \leq 1$). When t gets larger, Lemma 6.20 tells us that $P(M_t)$ is very small and so the sum doesn't get big in that case either. We will choose a way to split the sum so that this second part of the sum is bounded by a constant. In particular we split the sum up by

$$\sum_{t=0}^{n} P(M_t)t \leq \sum_{t=0}^{\lfloor 5\log n/\log\log n\rfloor} P(M_t)t + \sum_{t=\lceil 5\log n/\log\log n\rceil}^{n} P(M_t)t \tag{6.35}$$

For the sum over the smaller values of t, we just observe that in each term $t \leq 5\log n/\log\log n$ so that

$$\begin{aligned}
\sum_{t=0}^{5\log n/\log\log n} P(M_t)t &\leq \sum_{t=0}^{5\log n/\log\log n} P(M_t)5\log n/\log\log n && (6.36)\\
&= 5\log n/\log\log n \sum_{t=0}^{5\log n/\log\log n} P(M_t) && (6.37)\\
&\leq 5\log n/\log\log n && (6.38)
\end{aligned}$$

(Note that we are not using Lemma 6.20 here; only the fact that the probabilities of disjoint events cannot add to more than 1.) For the rightmost sum in Equation 6.35, we want to first compute an upper bound on $P(M_t)$ for $t = (5\log n/\log\log n)$. Using Lemma 6.20, and doing a bit of calculation we get that in this case $P(M_t) \leq 1/n^2$. Since the bound on $P(M_t)$ from Lemma 6.20 decreases as t grows, and $t \leq n$, we can bound the right sum by

$$\sum_{t=5\log n/\log\log n}^{n} P(M_t)t \leq \sum_{t=5\log n/\log\log n}^{n} \frac{1}{n^2}n \leq \sum_{t=5\log n/\log\log n}^{n} \frac{1}{n} \leq 1. \tag{6.39}$$

Combining Equations 6.38 and 6.39 with 6.35 we get the desired result.

Theorem 6.21 *If we hash n items into a hash table of size n, the expected maximum list length is $O(\log n/\log\log n)$.*

The choice to break the sum into two pieces here—and especially the breakpoint we chose—may have seemed like magic. What is so special about $\log n / \log\log n$? Consider the bound on $P(M_t)$. If you asked what is the value of t for which the bound equals a certain value, say $1/n^2$, you get the equation $ne^t/t^t = n^{-2}$. If we try to solve the equation $ne^t/t^t = n^{-2}$ for t, we quickly see that we get a form that we do not know how to solve. (Try typing this into Mathematica or Maple, to see that it can't solve this equation either.) The equation we need to solve is somewhat similar to the simpler equation $t^t = n$. While this equation does not have a closed form solution, one can show that the t that satisfies this equation is roughly $c \log n / \log\log n$, for some constant c. This is why some multiple of $\log n / \log\log n$ made sense to try as the the magic value. For values much less than $\log n / \log\log n$ the bound provided on $P(M_t)$ is fairly large. Once we get past $\log n / \log\log n$, however, the bound on $P(M_t)$ starts to get significantly smaller. The factor of 5 was chosen by experimentation to make the second sum come out to be less than 1. We could have chosen any number between 4 and 5 to get the same result; or we could have chosen 4 and the second sum would have grown no faster than the first.

Important Concepts, Formulas, and Theorems

1. *Expected Number of Keys per Slot in Hash Table.* In hashing n items into a hash table of size k, the expected number of items that hash to any one location is n/k.

2. Expected Number of Empty Slots in Hash Table. In hashing n items into a hash table with k locations, the expected number of empty locations is $k(1 - \frac{1}{k})^n$.

3. Collision in Hashing. We say that we have a *collision* when we hash an item to a location that already contains an item.

4. *The Expected Number of Collisions in Hashing.* In hashing n items into a hash table with k locations, the expected number of collisions is $n - k + k(1 - \frac{1}{k})^n$.

5. *Harmonic Number.* The quantity $\sum_{i=1}^{k} \frac{1}{i}$ is known as a *harmonic number*, and is sometimes denoted by H_k. It is a fact that that $\sum_{i=1}^{k} \frac{1}{i} = \Theta(\log k)$, and more precisely

$$\frac{1}{2} + \ln k \leq H_k \leq 1 + \ln k.$$

6. *Euler's Constant.* As n gets large, $H_n - \ln n$ approaches a limit called *Euler's constant*; Euler's constant is about .58.

7. *Expected Number of Hashes until all Slots of a Hash Table Are Occupied.* The expected number of items needed to fill all slots of a hash table of size k is between $k \ln k + \frac{1}{2}k$ and $k \ln k + k$.

8. *Expected Maximum Number of Keys per Slot.* If we hash n items into a hash table of size n, the expected maximum list length is $O(\log n / \log\log n)$.

Problems

1. A candy machine in a school has d different kinds of candy. Assume (for simplicity) that all these kinds of candy are equally popular and there is a large supply of each. Suppose that c children come to the machine and each purchases one package of candy. One of the kinds of candy is a Snackers bar. What is the probability that any given child purchases a Snackers bar? Let Y_i be the number of Snackers bars that child i purchases, so that Y_i is either 0 or 1. What is the expected value of Y_i? Let Y be the random variable $Y_1 + Y_2 + \cdots + Y_c$. What is the expected value of Y? What is the expected number of Snackers bars that is purchased? Does the same result apply to any of the varieties of candy?

2. Again as in the previous exercise, we have c children choosing from among ample supplies of d different kinds of candy, one package for each child, and all choices equally likely. What is the probability that a given variety of candy is chosen by no child? What is the expected number of kinds of candy chosen by no child? Suppose now that $c = d$. What happens to the expected number of kinds of candy chosen by no child?

3. How many children do we expect to have to observe buying candy until someone has bought a Snackers bar?

4. How many children to we expect to have to observe buying candy until each type of candy has been selected at least once?

5. If we have 20 kinds of candy, how many children have to buy candy in order for the probability to be at least one half that (at least) two children buy the same kind of candy?

6. What is the expected number of duplications among all the candy the children have selected?

7. Compute the values on the left-hand and right-hand side of the inequality in Lemma 6.18 for $n = 2$, $t = 0, 1, 2$ and for $n = 3$, $t = 0, 1, 2, 3$.

8. When we hash n items into k locations, what is the probability that all n items hash to different locations? What is the probability that the ith item is the first collision? What is the expected number of items we must hash until the first collision? Use a computer program or spreadsheet to compute the expected number of items hashed into a hash table until the first collision with $k = 20$ and with $k = 100$.

9. We have seen a number of occasions when our intuition about expected values or probability in general fails us. When we wrote down Equation 6.26 we said that the expected number of occupied locations is k minus the expected number of unoccupied locations. While this seems obvious, there is a short proof. Give the proof.

10. Write a computer program that prints out a table of values of the expected number of collisions with n keys hashed into a table with k locations for interesting values of n and k. Does this value vary much as n and k change?

11. Suppose you hash n items into a hash table of size k. It is natural to ask about the time it takes to find an item in the hash table. We can divide this into two cases, one when the item is not in the hash table (an unsuccessful search), and one when the item is in the hash table (a successful search). Consider first the unsuccessful search. Assume the keys hashing

to the same location are stored in a list with the most recent arrival at the beginning of the list. Use our expected list length to bound the expected time for an unsuccessful search. Next consider the successful search. Recall that when we insert items into a hash table, we typically insert them at the beginning of a list, and so the time for a successful search for item i should depend on how many entries were inserted after item i. Carefully compute the expected running time for a successful search. Assume that the item you are searching for is randomly chosen from among the items already in the table. (Hint: The unsuccessful search should take roughly twice as long as the successful one. Be sure to explain why this is the case.)

12. Suppose I hash $n \log n$ items into n buckets. What is the expected maximum number of items in a bucket?

13. The fact that $\lim_{n\to\infty}(1+\frac{1}{n})^n = e$ (where n varies over integers) is a consequence of the fact that $\lim_{h\to 0}(1+h)^{\frac{1}{h}} = e$ (where h varies over real numbers). Thus if h varies over negative real numbers, but approaches 0, the limit still exists and equals e. What does this tell you about $\lim_{n\to-\infty}(1+\frac{1}{n})^n$? Using this and rewriting $(1-\frac{1}{n})^n$ as $(1+\frac{1}{-n})^n$ show that

14. What is the expected number of empty slots when we hash $2k$ items into a hash table with k slots? What is the expected fraction of empty slots close to when k is reasonably large?

15. Using whatever methods you like (hand calculations or computer), give upper and/or lower bounds on the value of the x satisfying $x^x = n$.

16. Professor Max Weinberger decides that the method proposed for computing the maximum list size is much too complicated. He proposes the following solution. Let X_i be the size of list i. Then what we want to compute is $E(\max_i(X_i))$. Well

$$E(\max_i(X_i)) = \max_i(E(X_i)) = \max_i(1) = 1.$$

What is the flaw in his solution?

17. Prove as tight upper and lower bounds as you can on $\sum_{i=1}^{k}\frac{1}{i}$. For this purpose it is useful to remember the definition of the natural logarithm as an integral involving $1/x$ and to draw rectangles and other geometric figures above and below the curve.

18. Notice that $\ln n! = \sum_{i=1}^{n}\ln i$. Sketch a careful graph of $y = \ln x$, and by drawing in geometric figures above and below the graph, show that

$$\sum_{i=1}^{n}\ln i - \frac{1}{2}\ln n \le \int_1^n \ln x\,dx \le \sum_{i=1}^{n}\ln i.$$

Based on your drawing, which inequality do you think is tighter? Use integration by parts to evaluate the integral. What bounds on $n!$ can you get from these inequalities? Which one do you think is tighter? How does it compare to Stirling's approximation? What big Oh bound can you get on $n!$?