Introduction to Algorithm Design

Annalise Tarhan

February 9, 2021

1 Show that $a + b$ can be less than $min(a, b)$.

$a = -5 \quad b = -10 \quad a + b = -15 \quad min(a, b) = -10$

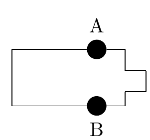
2 Show that $a \times b$ can be less than $min(a, b)$.

$a = -5 \quad b = 10 \quad a \times b = -50 \quad min(a, b) = -5$

3 Design/draw a road network with two points a and b such that the fastest route between a and b is not the shortest route.

This network requires two routes between a and b, one slightly longer than the other. The shorter route has a speed limit of 25 mph and the longer route has a speed limit of 45 mph.

4 Design/draw a road network with two points a and b such that the fastest route between a and b is not the route with the fewest turns.



5 The knapsack problem is as follows: given a set of integers $S = \{s_1, s_2, ..., s_n\}$, and a target number T, find a subset of S that adds up exactly to T. For each of the following algorithms, find a counterexample where the algorithm does not find a solution that leaves the knapsack completely full, even though a full-knapsack solution exists.

5.1 Put the elements of S in the knapsack in left to right order if they fit, that is, the first-fit algorithm.

$S = \{9, 10\} \quad T = 10$

5.2 Put the elements of S in the knapsack from smallest to largest, that is, the best-fit algorithm.

$S = \{1, 3, 4\} \quad T = 7$

5.3 Put the elements of S in the knapsack from largest to smallest.

$S = \{3, 5, 7\} \quad T = 8$

6 The set cover problem is as follows: given a set S of subsets $\{S_1, S_2, ..., S_m\}$ of the universal set $U = \{1, ..., n\}$, find the smallest subset of subsets $T \subseteq S$ such that $\cup_{t_i \in T} t_i = U$. Find a counterexample for the following algorithm: Select the largest subset for the cover, and then delete all its elements from the universal set. Repeat until all elements are covered.

$S = \{S_1 = \{1, 3, 5\}, S_2 = \{1, 2, 4, 5\}, S_3\{2, 4, 6\}\}$
S_1 and S_3 are sufficient to cover $\{1, ..., 6\}$, but this algorithm would choose S_2 first, requiring it to include all three subsets.

7 The maximum clique problem in a graph $G = (V, E)$ asks for the largest subset C of vertices V such that there is an edge in E between every pair of vertices in C. Find a counterexample for the following algorithm: Sort the vertices of G from highest to lowest degree. Considering the vertices in order of degree, for each vertex add it to the clique if it is a neighbor of all vertices currently in the clique. Repeat until all vertices have been considered.

Consider a graph where every vertex is has an edge to every other vertex. Then, add another vertex Z and connect in only with vertex A. Then, A will have the highest degree and the algorithm will consider it first. But, since no other vertex has an edge to Z, it will return $\{A\}$ instead of the optimal solution:$\{A, ..., Y\}$.

8 Prove the correctness of the following recursive algorithm to multiply two natural numbers, for all integer constants $c \geq 2$.

```
Multiply (y, z)
    if z=0 then return(0) else
    return(Multiply (cy, ⌊z/c⌋) + y * (z mod c))
```

Base case: $z = 0$ The algorithm immediately, and correctly, returns 0, since $y * 0 = 0$ for all values of y.

Inductive step: Assume the algorithm is correct for all values of z such that $0 < z \leq n - 1$. Now, show it is correct for $z = n$. Since $n \neq 0$, the algorithm returns the sum of $Multiply(cy, \lfloor n/c \rfloor)$ and $y * (z \ mod \ c))$. Let $n = ac + b$ for some natural number values of a and b. Then, $\lfloor n/c \rfloor = a$. Since $c \geq 2$ and n is a natural number, $0 \leq \lfloor n/c \rfloor) < n$, so by our assumption, $Multiply$ will return the correct result of $cy * a$. The second addend, $y * (n \ mod \ c)$ equals $y * b$. The algorithm then returns $cy * a + y * b = y(ac + b) = y * n$, the correct answer.

9 Prove the correctness of the following algorithm for evaluating a polynomial $a_nx^n + a_{n-1}x^{n-1} + ... + a_1x + a_0$.

```
Horner(a, x)
   p = a_n
   for i from n − 1 to 0
      p = p ∗ x + a_i
   return p
```

Base case: $n = 0$ Here, p is assigned a_0, and the for loop doesn't execute, since $n-1 = -1$ and this for loop seems to decrement i from the first argument to 0. Finally, it returns $p = a_0$, the correct answer.

Inductive step: Assume the algorithm is correct for $n-1$, returning $a_{n-1}x^{n-1} + ... + a_1x + a_0$. Now, prove it holds for n. The algorithm will initially assign a_n to p. Then, for i from $n-1$ to 0, it resets p to equal $p * x + a_i$. In the first step, p is reset to $a_nx + a_{n-1}$. From there, the for loop continues exactly as for $n-1$, except instead of starting with $p = a_{n-1}$, it starts with $p = a_nx + a_{n-1}$. Since the return value for $n-1$ is $a_{n-1}x^{n-1} + ... + a_1x + a_0$ and we assume it to be correct, substitute $a_nx + a_{n-1}$ for a_{n-1}. This gives $(a_nx + a_{n-1})x^{n-1} + ... + a_1x + a_0$, which simplifies to $a_nx^n + a_{n-1}x^{n-1} + ... + a_1 + a_0$, the correct answer.

10 Prove the correctness of the following sorting algorithm.

```
Bubblesort(A)
   for i from n to 1
      for j from 1 to i − 1
         if (A[j] > A[j + 1])
            swap the values of A[j] and A[j + 1]
```

Base case: The size of A, and therefore n, is 0. The outer loop iterates once, with i initially set to 0 and then to 1. The inner loop would iterate from 1 to 0, but $1 > 0$, so it doesn't execute at all. A remains unchanged, which is the expected behavior, since a list with only one element is by definition already sorted.

Iterative step: Assume that for A of size $n-1$, the algorithm is correct, and the array is sorted correctly. Prove that it is also correct for A of size n. Adding another element to A will cause the outer loop to begin with $i = n$ instead of $i = n - 1$. This will cause the inner loop, which starts at the beginning of the array and compares each neighboring pair of elements, to compare the first bubbled element, which is guaranteed by the assumption to be the largest of the first $n-1$ elements, to the final nth element. If the bubbled element is larger, it will take its place and the first $n-1$ elements will be sorted as before. If the nth element is largest, it will remain in place and the sorting will also be sorted as before. In either case, A is sorted correctly.

11 The greatest common divisor of positive integers x and y is the largest integer d such that d divides x and d divides y. Euclid's algorithm to compute $gcd(x, y)$ where $x > y$ reduces the task to a smaller problem: $gcd(x, y) = gcd(y, x \ mod \ y)$. Prove that Euclid's algorithm is correct.

Let z be the greatest common divisor of x and y. Then $x = mz$, $y = nz$, and $x = ay + b$ where $b = x \ mod \ y$ and $a, b, n, m \in \mathbb{N}$. Then $mz = anz + b$, which leads to $b = z(m - an)$. Since a, n, and m are natural numbers, $m - an$ must be as well, so z must also divide b. Thus, z is a common divisor of y and $x \ mod \ y$.

By contradiction, assume there is a number greater than z that also divides b and y. But since $x = ay + b$, it would divide both ay and b, and therefore x. This contradicts the given that z is the greatest common divisor of x and y, so $gcd(x, y) = gcd(y, \ x \ mod \ y)$

12 Prove that $\sum_{i=1}^{n} i = n(n + 1)/2$ for $n \geq 0$, by induction.

Base case: $n = 0$ $\quad \sum_{i=1}^{0} = 0$ and $0(0 + 1)/2 = 0$

Inductive step: Assume $\sum_{i=1}^{n-1} i = (n - 1)(n)/2$.
Prove $\sum_{i=1}^{n} i = n(n + 1)/2$.

Left hand side: $\sum_{i=1}^{n} i = n + \sum_{i=1}^{n-1} i$
Right hand side:
$n(n + 1)/2 = n - 2n/2 + n(n + 1)/2$
$n + \frac{n(n+1)-2n}{2}$
$n + \frac{n^2+n-2n}{2}$
$n + \frac{n(n-1)}{2}$
Thus, proving that $\sum_{i=1}^{n} i = n(n + 1)/2$ is equivalent to proving $n + \sum_{i=1}^{n-1} i = n + \frac{n(n-1)}{2}$. Since we can assume that $\sum_{i=1}^{n-1} i = (n - 1)(n)/2$ and since $n = n$, it is indeed true.

13 Prove that $\sum_{i=1}^{n} i^2 = n(n{+}1)(2n{+}1)/6$ for $n \geq 0$, by induction.

Base case: $n = 0$ $\quad \sum_{i=1}^{0} i^2 = 0 = 0(1)(1)/6$

Inductive step: Assume $\sum_{i=1}^{n-1} i^2 = (n-1)(n)(2n-1)/6$.
Prove $\sum_{i=1}^{n} i^2 = n(n+1)(2n+1)/6$.

Left hand side:
$\sum_{i=1}^{n} i^2 = n^2 + \sum_{i=1}^{n-1} i^2$
Right hand side:
$n(n+1)(2n+1)/6$
$n^2 + n(n+1)(2n+1) - 6n^2/6$
$n^2 + (2n^3 - 3n^2 + n)/6$
$n^2 + (n-1)(n)(2n-1)/6$.

Since $\sum_{i=1}^{n-1} i^2 = (n-1)(n)(2n-1)/6$,
we know that $n^2 + \sum_{i=1}^{n-1} i^2 = n^2 + (n-1)(n)(2n-1)/6$,
so $\sum_{i=1}^{n} i^2 = n(n+1)(2n+1)/6$.

14 Prove that $\sum_{i=1}^{n} i^3 = n^2(n+1)^2/4$ for $n \geq 0$, by induction.

Base case: $n = 0$ $\quad \sum_{i=1}^{0} i^3 = 0 = 0^2 1^2/4$

Inductive step: Assume $\sum_{i=1}^{n-1} i^3 = (n-1)^2 n^2/4$.
Prove $\sum_{i=1}^{n} i^3 = n^2(n+1)^2/4$.

Left hand side:
$\sum_{i=1}^{n} i^3 = n^3 + \sum_{i=1}^{n-1} i^3$
Right hand side:
$n^2(n+1)^2/4$
$n^3 + (n^2(n+1)^2 - 4n^3)/4$
$n^3 + (n^4 - 2n^3 + n^2)/4$
$n^3 + (n-1)^2 n^2/4$

Since $\sum_{i=1}^{n-1} i^3 = (n-1)^2 n^2/4$,
we know that $n^3 + \sum_{i=1}^{n-1} i^3 = n^3 + (n-1)^2 n^2/4$,
so $\sum_{i=1}^{n} i^3 = n^2(n+1)^2/4$.

15 Prove that $\sum_{i=1}^{n} i(i+1)(i+2) = n(n+1)(n+2)(n+3)/4$.

Base case: $n = 0$ $\quad \sum_{i=1}^{0} i(i+1)(i+2) = 0 = 0(1)(2)(3)/4$

Inductive step: Assume $\sum_{i=1}^{n-1} i(i+1)(i+2) = (n-1)(n)(n+1)(n+2)/4$.
Prove $\sum_{i=1}^{n} i(i+1)(i+2) = n(n+1)(n+2)(n+3)/4$.

Left hand side:
$\sum_{i=1}^{n} i(i+1)(i+2) = n(n+1)(n+2) + \sum_{i=1}^{n-1} i(i+1)(i+2)$

Right hand side:
$n(n+1)(n+2)(n+3)/4$
$n(n+1)(n+2) + \frac{n(n+1)(n+2)(n+3)}{4} - \frac{4n(n+1)(n+2}{4}$
$n(n+1)(n+2) + (n^4 + 2n^3 - n^2 - 2n)/4$
$n(n+1)(n+2) + (n-1)n(n+1)(n+2)/4$

Since $\sum_{i=1}^{n-1} i(i+1)(i+2) = (n-1)(n)(n+1)(n+2)/4$, we know that $n(n+1)(n+2)+\sum_{i=1}^{n-1} i(i+1)(i+2) = n(n+1)(n+2)+(n-1)(n)(n+1)(n+2)/4$, so $\sum_{i=1}^{n} i(i+1)(i+2) = n(n+1)(n+2)(n+3)/4$.

16 Prove by induction on $n \geq 1$ that for every $a \neq 1$, $\sum_{i=0}^{n} a^i = \frac{a^{n+1}-1}{a-1}$.

Base case: $n = 1$ $\quad \sum_{i=0}^{1} a^i = a + 1 = \frac{a^2-1}{a-1} (= \frac{(a+1)(a-2)}{a-1})$

Inductive step: Assume $\sum_{i=0}^{n-1} a^i = \frac{a^n-1}{a-1}$. Prove $\sum_{i=0}^{n} a^i = \frac{a^{n+1}-1}{a-1}$.

Left hand side:
$\sum_{i=0}^{n} a^i = a^n + \sum_{i=0}^{n-1} a^i$

Right hand side:
$\frac{a^{n+1}-1}{a-1}$
$a^n + \frac{a^{n+1}-1}{a-1} - \frac{a^n(a-1)}{a-1}$
$a^n + \frac{a^{n+1}-1-a^{n+1}+a^n)}{a-1}$
$a^n + \frac{a^n-1}{a-1}$

Since $\sum_{i=0}^{n-1} a^i = \frac{a^n-1}{a-1}$, we know that $a^n + \sum_{i=0}^{n-1} a^i = a^n + \frac{a^n-1}{a-1}$, so $\sum_{i=0}^{n} a^i = \frac{a^{n+1}-1}{a-1}$ for all $n \geq 1$ and $a \neq 1$.

17 Prove by induction that for $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i(i+1)} = \frac{n}{n+1}$.

Base case: $n = 1$ $\quad \sum_{i=1}^{1} \frac{1}{i(i+1)} = 1/2 = \frac{1}{1+1}$

Inductive step: Assume $\sum_{i=1}^{n-1} \frac{1}{i(i+1)} = \frac{n-1}{n}$. Prove $\sum_{i=1}^{n} \frac{1}{i(i+1)} = \frac{n}{n+1}$.

Left hand side:
$\sum_{i=1}^{n} \frac{1}{i(i+1)} = \frac{1}{n(n+1)} + \sum_{i=1}^{n-1} \frac{1}{i(i+1)}$

Right hand side:
$\frac{n}{n+1}$
$\frac{1}{n(n+1)} + \frac{n}{n+1} - \frac{1}{n(n+1)}$
$\frac{1}{n(n+1)} + \frac{n^2-1}{n(n+1)}$
$\frac{1}{n(n+1)} + \frac{n-1}{n}$

Since $\sum_{i=1}^{n-1} \frac{1}{i(i+1)} = \frac{n-1}{n}$, we know that $\frac{1}{n(n+1)} + \sum_{i=1}^{n-1} \frac{1}{i(i+1)} = \frac{1}{n(n+1)} + \frac{n-1}{n}$, so $\sum_{i=1}^{n} \frac{1}{i(i+1)} = \frac{n}{n+1}$ for all $n \geq 1$.

18 Prove by induction that $n^3 + 2n$ is divisible by 3 for all $n \geq 0$.

Base case: $n = 0$ $\quad 0^3 + 2(0) = 0$ and $0/3 = 0$.

Inductive step: Assume $(n-1)^3 + 2(n-1) = 3m$ for some $m \in \mathbb{N}$. Prove $n^3 + 2n = 3m'$ for some $m' \in \mathbb{N}$.

Starting from the assumption,
$(n-1)^3 + 2(n-1) = 3m$
$n^3 - 3n^2 + 3n - 1 + 2n - 2 = 3m$
$(n^3 + 2n) + (-3n^2 + 3n - 3) = 3m$
$n^3 + 2n = 3m + 3n^2 - 3n + 3$
$n^3 + 2n = 3(m + n^2 - n + 1)$
Since $m' = m + n^2 - n + 1 \in \mathbb{N}$, $n^3 + 2n$ is divisible by 3 for all $n \geq 0$.

19 Prove by induction that a tree with n vertices has exactly $n - 1$ edges.

Base case: $n = 1$ $\quad$ There is only one vertex, so there is no other vertex with which it can share an edge. So, it has $1 - 1 = 0$ edges.

Inductive step: Assume a tree with $n - 1$ vertices has $n - 2$ edges. Prove that a tree with n vertices has $n-1$ edges. Beginning with the $n-1$ vertex tree, add another vertex. It must have exactly one edge to one parent. This gives a tree with n vertices and $n-1$ edges.

20 Prove by induction that the sum of the cubes of the first n positive integers is equal to the square of the sum of these integers, that is, $\sum_{i=1}^{n} i^3 = (\sum_{i=1}^{n} i)^2$.

Base case: $n = 1$ $\sum_{i=1}^{1} i^3 = 1 = (\sum_{i=1}^{1} i)^2$.

Inductive step: Assume $\sum_{i=1}^{n-1} i^3 = (\sum_{i=1}^{n-1} i)^2$. Prove $\sum_{i=1}^{n} i^3 = (\sum_{i=1}^{n} i)^2$.

Left hand side:
$\sum_{i=1}^{n} i^3$
$n^3 + \sum_{i=1}^{n-1} i^3$

Right hand side:
$(\sum_{i=1}^{n} i)^2$
$(n + \sum_{i=1}^{n-1} i)^2$
$n^2 + 2n \sum_{i=1}^{n-1} i + (\sum_{i=1}^{n-1} i)^2$
The sum of the natural numbers from one to n is $n(n+1)/2$, so
$\sum_{i=1}^{n-1} i = n(n-1)/2$
$n^2 + 2n(n(n-1)/2) + (\sum_{i=1}^{n-1} i)^2$
$n^2 + n^3 - n^2 + (\sum_{i=1}^{n-1} i)^2$
$n^3 + (\sum_{i=1}^{n-1} i)^2$

Since $\sum_{i=1}^{n-1} i^3 = (\sum_{i=1}^{n-1} i)^2$, we know that $n^3 + \sum_{i=1}^{n-1} i^3 = n^3 + (\sum_{i=1}^{n-1} i)^2$, so $\sum_{i=1}^{n} i^3 = (\sum_{i=1}^{n} i)^2$ for all positive integers n.

21 Do all the books you own total at least one million pages? How many total pages are stored in your school library?

I have about a dozen CS textbooks with around 800 pages each. At around 10,000 pages total, this comes nowhere near a million pages. All I remember about my school library is that it was enormous. Estimate 300 racks of books across several floors. Each rack has two sides and, let's say, five sections. Each section had five shelves with around fifty books each. Since most of the books were not textbooks, let's say they were on average 400 pages each. $400 * 50 *$

$5 * 5 * 2 * 300 = 300,000,000$ pages. And that's just in one of Berkeley's many libraries!

22 How many words are there in this textbook?

A typical textbook might have 50 lines of text per page, but this book contains code, images, and math. Assume 30 lines of text per page, with 10 words each. At almost 800 pages, this gives $800 * 10 * 30 = 240,000$ words.

23 How many hours are in one million seconds? How many days?

$1,000,000/60/60 = 1,000,000/3600$ is a bit more than $1,000,000/4000 = 250$, so estimate 275 hours. To estimate days, divide 275/24, which is just a tiny bit more than $275/25 = 11$, so estimate 12 days.

24 Estimate how many cities and towns there are in the United States.

The population of the United States is around 330,000,000. Estimate four fifths live in cities and one fifth live in towns. If cities range in population from 50,000 to several million, put the average at one million. Let the average population of a town be 25,000. Then $330,000,000 * (4/5)/1,000,000 = 264$ cities and $330,000,000 * (1/5)/25,000 = 2,640$ towns.

25 Estimate how many cubic miles of water flow out of the mouth of the Mississippi River each day. Do not look up any supplemental facts. Describe all assumptions you made in arriving at your answer.

Let's guess that the mouth of the Mississippi river is two miles wide and 100 feet deep. If water is flowing at a rate of 10 miles per hour, that gives $10*2*100/5280$, about .38 cubic miles per hour. Per day, that's $.38 * 24$, around 9 cubic miles per day.

26 How many Starbucks or McDonalds's locations are there in your country?

The vast majority of Starbucks locations in Turkey are going to be in the handful of large cities, with just a few dozen scattered around the rest of the country. Estimate three dozen in Istanbul, two dozen in Ankara, and a dozen each in a few other cities. Total, I'd estimate around 100 locations.

27 How long would it take to empty a bathtub with a drinking straw?

If a bathtub is 4'x2.5'x2', it contains 20 cubic feet of water, or around 35,000 cubic inches of water. Assume a drinking straw can hold a third of a cubic inch of water, and that you can empty a straw's worth of water every three seconds. Then it will take 350,000 seconds, 60,000 minutes, or 1,000 hours to empty the bathtub.

28 Is disk drive access time usually measured in milliseconds or microseconds? Does your RAM memory access a word in more or less than a microsecond? How many instructions can your CPU execute in one year if the machine is left running all the time?

It is usually measured in milliseconds, thousandths of a second. RAM access is much faster, accessing a word in less than a microsecond. If a CPU is left running for a year, virtually all of the instructions will have to be accessed from disk memory. If an instruction takes 20 milliseconds to access, that's 50 instructions per second, 300 per minute, 1800 per hour, 43,200 per day, and over 15 million in a year. This assumes the instructions are not sequential, which would be unfortunate.

29 A sorting algorithm takes 1 second to sort 1,000 items on your machine. How long will it take to sort 10,000 items...

29.1 if you believe that the algorithm takes time proportional to n^2?

If the number of items increases by $n = 10$, the time will increase by $n^2 = 10^2 = 100$, so the algorithm will take 100 seconds.

29.2 if you believe that the algorithm takes time roughly proportional to $n\ log\ n$?

10 *log* 10 is roughly 33, so the algorithm will take just over 30 seconds.

30 Implement the two TSP heuristics of Section 1.1. Which of them gives better solutions in practice? Can you devise a heuristic that works better than both of them?

In practice, closest pair was 12% faster than nearest neighbor. I was going to suggest that a heuristic that works better than both is running each of them and choosing the better result, but after tens of thousands of simulations, not once did the nearest neighbor algorithm yield a shorter route. So no, I can't devise a better heuristic than closest pair.

31 Describe how to test whether a given set of tickets establishes sufficient coverage in the Lotto problem of Section 1.8. Write a program to find good ticket sets.

I've chosen to define "good ticket sets" very, very loosely. The program chooses tickets randomly until it finds a covering set, then deletes any that are unnecessary. But, we were given an example of five tickets that cover all the possibilities. Mine takes 25-30. The process for testing if a ticket set establishes coverage is this: For each of the $\binom{15}{4} = 1365$ combinations of winning numbers, check that at least one of the tickets in the set has at least three of them.

32 Write a function to perform integer division without using either the / or * operators. Find a fast way to do it.

This approach initializes an accumulator to 0 and a counter to -1, then repeatedly adds the divisor to the accumulator and increments the counter until the accumulator is larger than the dividend. The counter's value is the quotient. (Negative numbers are handled appropriately.) A faster approach would use bit shifting and the speedup would be especially significant for large dividends with small divisors.

```
int divide(int divisor, int dividend) {
        if (dividend == 0) return 0;
        if (divisor == 0) return NAN;   // Dividing by zero
        int or, end, negative_flag;     // Negative input
        if (divisor < 0 && dividend < 0) {
                or = 0-divisor;
                end = 0-dividend;
                negative_flag = 0;
        } else if (divisor < 0) {
                or = 0-divisor;
                end = dividend;
                negative_flag = 1;
        } else if (dividend < 0) {
                or = divisor;
                end = 0-dividend;
                negative_flag = 1;
        } else {
                or = divisor;
                end = dividend;
                negative_flag = 0;
        }
        if (or > end) return 0;
        int count = -1;
        int accumulator = 0;
        while (accumulator <= end) {
                accumulator += or;
                count++;
        }
        if (negative_flag) {
                return 0-count;
        } else {
                return count;
        }
}
```

33 There are twenty-five horses. At most, five horses can race together at a time. You must determine the fastest, second fastest, and third fastest horses. Find the minimum number of races in which this can be done.

First, race all the horses in groups of five. We can safely ignore the slowest two in each race. Then, race the five winners. Not only the slowest two in this race, but the other horses from the their original race groups can now be ignored as well. This leaves nine horses after six races. Let A, B, and C be the three fastest horses in that sixth race. A' and A" are the two next fastest from A's original group, and likewise with B', B", C', and C":
A A' A"
B B' B"
C C' C"
We know that A must be the fastest horse and that A' or B must be the second fastest. If A' turns out to be faster than B, the third fastest is either A" or B. If B turns out to be faster, the third fastest could be A', B', or C. This will only take one race to determine: A', A", B, B', and C. Altogether, this takes seven races.

34 How many piano tuners are there in the entire world?

The number of piano tuners is a function of the number of pianos and piano ownership density. If there are only a handful of pianos in a given area, it is unlikely a piano tuner in the area could make a living. Since piano ownership is relatively expensive and only traditional in Western cultures, we can assume that piano ownership is only high enough to support piano tuners in medium sized or larger cities in North America and Europe, and large cities in the rest of the world. Considering only those cities, suppose 1% of households in NA/E and .1% elsewhere own acoustic pianos, that pianos are tuned once a year, and that a piano tuner services three pianos per workday. If we let household size equal three people in NA/E, four people elsewhere, and workdays per year equal 260, the formula then becomes:
(Pianos in need of yearly tuning) / (Tunings a tuner can perform per year)
(Population of medium-large NA/E cities)(1/3)(.01)/(3*260)
+ (Population of large cities outside NA/E)(1/4)(.001)/(3*260)

35 How many gas stations are there in the United States?

Assume two thirds of Americans between 16 and 80 own cars and that they fill up their tanks once per week. Some gas stations will be far busier than others, but assume that between 8 am and 8 pm, half service one car per minute and half service one car every ten minutes. On average, each services one every 5.5 minutes for 131 total per day. The number of gas stations in the US equals the number of fill ups per week divided by the number each station can do in a week.

(American 16-80 year olds * 2/3) / (131 fill ups per day * 7 days)

36 How much does the ice in a hockey rink weigh?

Hockey rinks are 200 feet long by 85 feet wide, or 17,000 square feet, and the ice is approximately 3/4" thick. This gives 12,750 cubic feet of ice. Ice weighs about 57.4 pounds per cubic foot, so the total weight is approximately 731,850 pounds.

37 How many miles of road are there in the United States?

What immediately springs to mind is the network of highways crisscrossing the country, but even though they feature prominently on road maps, the tens of thousands of miles of major highways are dwarfed by the densely packed residential and commercial roads everywhere people live and work. Much of the United States is empty land and much of the rest is sparsely populated, so assume that a third of the country, by surface area, is covered in roads. For simplicity, assume that the roads in this road-ed part of the country are laid out in a grid with blocks 1/4 mile square. In one square mile, this gives four horizontal and four vertical roads of one mile each, for eight miles of road per square mile. The total miles of road is then: Surface area of US * 1/3 * 8

38 On average, how many times would you have to flip open the Manhattan phone book at random in order to find a specific name?

When the probability of one trial succeeding is $1/p$, the expected number of trials for one success is p. The probability of randomly choosing the right page is one divided by the number of pages in the phone book, so what we are really trying to discover is how many pages there are in a Manhattan phone book. The number of pages depends on the population and the layout of the book. The number of times you'd have to flip the phone book open, then, is the population of Manhattan divided by the number of listing per page in the phone book.