

Warm-up

What is the greatest number you can express in 10 seconds?

You may only use numerical digits and mathematical operations, not references or indirection; *e.g.*, you may not write something like, "the largest number written by anyone else in the room +1."

Warm-up

How is Computer Science different from Software Development?

Structures: Discrete and Continuous

In this class we will discuss, not surprisingly, discrete structures, but what are discrete structures? First, let's get out of the way the idea of a *set*. Without giving it too much thought, we start with the following:

Definition 1. A set is an unordered collection of unique *elements*.

We will denote the contents of a set using curly braces, $\{\}$. So for instance, the following are sets:

$$\{A, B, C\}$$

The real numbers, \mathbb{R}

$$\{\text{Tim, an elm tree, } \mathbb{R}\}$$

$$\{0.4x \mid x \text{ is an odd, positive integer}\} = \{0.41, 0.43, 0.45, \dots\}$$

The last example is given in *set builder notation*, which we will make copious use of through the course. The idea is to give the elements of the set in terms of variables, then to give the constraints of the variables.

Notice that the first and third examples have a finite number of elements. The number of elements of a set are called its *cardinality*, so that the first set has a cardinality of 3, where we would write

$$|\{A, B, C\}| = 3.$$

The second and fourth examples have infinite cardinality.

It is also a good time to mention that if S is a set, and X is another set that contains elements from the set S , then X is a subset of S , written $X \subset S$. We have

$$\{A, B\} \subset \{A, B, C\}$$

$$\mathbb{Q} \subset \mathbb{R}$$

$$S \subset S \text{ for any set } S$$

$$\emptyset \subset S \text{ for any set } S$$

We are now more or less in a position to say that discrete structures are those that are representable either by a finite set, or an infinite set that can be put into correspondence with the natural (counting) numbers (more about that in an upcoming lecture).

Practically speaking, we can count the elements of a discrete structure, where we cannot do so with a non-discrete structure. The real numbers, \mathbb{R} , are an example of a continuous structure, which is certainly not discrete (again, more on that soon).

Counting: The Sum Principle

Example 1 How many times is the comparison $A[i] > A[j]$ made in Line 3 below?

Example 1

```
1    for i = 1 to n-1
2        for j = i+1 to n
3            if (A[i] > A[j])
4                swap A[i] and A[j]
```

Counting: The Sum Principle

Example 1 How many times is the comparison $A[i] > A[j]$ made in Line 3 below?

Example 1

```

1      for i = 1 to n-1
2          for j = i+1 to n
3              if (A[i] > A[j])
4                  swap A[i] and A[j]
```

The solution is quite simple. The first time through the inner loop, we perform $n - 1$ comparisons. The second time through, we perform $n - 2$ comparisons. The i th time, we perform $n - i$ comparisons. So we get

$$(n - 1) + (n - 2) + \cdots + 1 = \frac{n(n - 1)}{2}$$

total comparisons, where we will prove this formula later.

The solution to this problem isn't nearly so useful as the reasoning that leads to it. For each i , let S_i be the set of comparisons made for that value of i . So S_1 contains the comparisons made when $i = 1$, S_2 contains the comparisons made when $i = 2$, and so on. Then it is important to note that

$$S_a \cap S_b = \emptyset$$

whenever $a \neq b$; i.e. the sets S_i are *disjoint*. If we let S be the set of all comparisons made for the entire code block, then

$$S = S_1 \cup S_2 \cup \cdots \cup S_{n-1} = \bigcup_{i=1}^{n-1} S_i$$

and the cardinality of S is given by

$$|S| = |S_1 \cup S_2 \cup \cdots \cup S_{n-1}| = |S_1| + |S_2| + \cdots + |S_{n-1}|.$$

This is important enough to note as a *proposition*, i.e., a useful, light-weight theorem:

Proposition 1. (Sum Principle) The size of a union of a family of mutually disjoint finite sets is the sum of the sizes of the sets.

We will prove this after a discussion on the principle of induction. We can also restate this proposition by making a useful definition.

If we can write the set S as the union of a family of non-empty, disjoint sets, $\mathcal{S} = \{S_1, \dots, S_n\}$, then we say that \mathcal{S} *partitions* S and that the *partition* \mathcal{S} is comprised of *blocks* S_1, \dots, S_n . We have

Proposition 2. (Sum Principle - Partition Form) If a finite set S has been partitioned into blocks, then the cardinality of S is the sum of the cardinality of the blocks.

Counting: The Product Principle

Example 2 How many multiplications (expressed in terms of r , m , and n) does this pseudocode carry out in total among all the iterations of Line 5?

Example 2

```
1      for i = 1 to r
2          for j = 1 to m
3              S = 0
4              for k = 1 to n
5                  S = S + A[i,k] * B[k,j]
6                  C[i,j] = S
```

Counting: The Product Principle

Example 2 How many multiplications (expressed in terms of r , m , and n) does this pseudocode carry out in total among all the iterations of Line 5?

Example 2

```
1      for i = 1 to r
2          for j = 1 to m
3              S = 0
4              for k = 1 to n
5                  S = S + A[i,k] * B[k,j]
6              C[i,j] = S
```

Counting explicitly, we see that the loop in Lines 4-5 always executes n times, giving n multiplications. The loop in Lines 2-5 executes m times, so that everytime it runs, we get nm multiplications. Finally, the outer loop, from Lines 1-5, executes r times, each time performing nm multiplications, for a total of nmr multiplications.

Again, the actual solution isn't as important as the reasoning used to obtain it. Why did we end up multiplying in this example instead of adding, as we did in the Example 1?

Let's walk through this example as we did before. For each j , let S_j be the set of multiplications performed. We get $|S_j| = n$ for every j . Now let T_i be the set of all multiplications performed for each index i . Then T_i is the union of the sets S_j . Using the Sum Principle, we get

$$|T_i| = \left| \bigcup_{j=1}^m S_j \right| = \sum_{j=1}^m |S_j| = \sum_{j=1}^m n = mn.$$

We now have a new proposition that acts as a shortcut under certain conditions for the Sum Principle:

Proposition 3. (Product Principle) The cardinality of a union of a family of mutually disjoint finite sets is the sum of the cardinalities of the sets.

The proof is the same as our reasoning above.

As it happens, we could have counted the comparisons from Example 1 using the Product Principle as well. Notice that for each unique pair, i and j , in Example 1, we make exactly one comparison. So we have that the number of comparisons is the same as the number of unique pairs, which, in turn, is the same as the number of two-element subsets of the set $\{1, 2, \dots, n\}$. We will discuss this later, but this is an example of a *bijection*. We now want to answer the question, in how many ways can we choose two elements from this set?

It should be relatively clear that there are n different ways to choose the first element. Once that element is chosen, it is gone, and we have $n - 1$ elements remaining for our second choice. You might think that the Product Principle tells us that there are $n(n - 1)$ ways to choose two elements from $\{1, 2, \dots, n\}$; however, what we have chosen are *ordered pairs*. Sets are *unordered* (go back and check the definition) so that choosing 2 then 5 is the same as choosing 5 then 2. Therefore $n(n - 1)$ gives us twice as many pairs as we need, and the correct result is given by $\frac{n(n-1)}{2}$.

This value comes up so often that it has its own name and notation. When we choose k unordered items from a set of cardinality n , we call it " n choose k " and write $\binom{n}{k}$. We have shown that

$$\binom{n}{k} = \frac{n(n-1)}{2}.$$

We will derive a formula for arbitrary intergers n and k with $n \geq k$ later.