

CSC215

Math and Computer Science



Recursion – Spock's Dilemma

- Spock's Dilemma
 - Kirk tells Spock that we have to head back to base after x amount of time. "There are n number of planets and we have time to visit k number of them."
 - Spock wants to know how many possible ways to arrange visiting k number of planets there are.
 - Order does not matter. So visiting x then y is the same as visiting y then x .

Recursion - Combinations

- Combinations

- How many ways are there to pick k items from a list of size n
- Interested in how many there are and not what they are.
- Commonly represented as $C(n,k)$.
- Best done iteratively.

$$\frac{n!}{k! * (n - k)!}$$

Example:

- Set of $n \{ 1, 2, 3 \}$
- How many ways are to order 2 numbers.
- $C(3,2)$
- $\{1,2\} , \{1,3\} , \{2,3\} = 3$

Example:

- Set of n { 1, 2, 3, 4 }
- How many ways are to order 3 numbers.
- $C(4,3)$
- {1,2,3} , {1,2,4}, {1,3,4}, {2,3,4} = 4

Example:

- Set of n { 1, 2, 3, 4 }
- How many ways are to order 2 numbers.
- $C(4,2)$
- {1,2} , {1,3} , {1,4}, {2,3}, {2,4}, {3,4} = 6

Breaking it down $C(4,2)$

- Set of n { 1, 2, 3, 4 }
- How many ways are to order 2 numbers.
- Let A = # of subsets that contain a 1 (just a # from the set)
 - {1,2} , {1,3} , {1,4}
 - $A = C(n-1,k-1) = C(3,1)$
- Let B = # of subsets that do not contain a 1. Set is now {2,3,4}
 - {2,3} , {2,4} , {3,4}
 - $B = C(n-1, k) = C(3,2)$
- Solution to $C(n,k) = A + B = C(3,1) + C(3,2)$

Base Cases for $C(n,k)$

$C(n,n)$ $k=n$	1 way to pick all of them
$C(n,0)$ $k=0$	1 way to pick nothing
$C(n,k)$ $k > n$	0 ways

Criteria for $C(n,k)$

$C(n,n)$ $k=n$	1 way to pick all of them
$C(n,0)$ $k=0$	1 way to pick nothing
$C(n,k)$ $k > n$	0 ways
$C(n,k)$	$C(n-1,k-1) + C(n-1,k)$

Writing the Code – Base Cases First

```
int choose( int n, int k)
{
    // base cases first
    if( n < k )
        return 0;
    if( n == k || k == 0 )
        return 1;
}
```

Writing the Code – Compute the Answer

```
int choose( int n, int k)
{
    int ans;
    // base cases first
    if( n < k )
        return 0;
    if( n == k || k == 0 )
        return 1;

    ans = choose(n-1,k-1) + choose( n-1, k );
    return ans;
}
```

Calling this Function

```
int main()
{  int numelements;
    int quantity;
    long result;

    cout << "Enter the number of elements in your set: ";
    cin >> numelements;
    cout << "Enter the number of element to choose from the set: ";
    cin >> quantity;
    cout << endl;
    result = choose( numelements, quantity );
    cout << "There are " << result << " combinations" << endl;
    return 0;
}
```