# Sorting and Shuffling Arrays

CS 233G

Intermediate C++ Programming for Games

# Linear Search

- You already know how to do this
- Start at the beginning and go through the array until you find what you're looking for
- For min or max, you have to process the whole array
- For a specific item, you stop when you find it

# Ordered or Sorted

- I'm thinking of a number between 0 and 255
- Let's play Hi/Lo. What's the number?
- That's right, 14
- A good way to guess this:
- 127,63,31,15,7,11,13,14
- Why is this a good way?
- That's right, eliminate ½ of the possabilites on each guess.

# Maximum Number of Steps

- (exponent Smallest power of 2 ) >= Num elements
- "Logarithmic" algorithm

| Array Elements | Max Comparisons |
|---|---|
| 256 = 2^8 | 8 |
| 65536 = 2^16 | 16 |
| 4294967296 = 2^32 | 32 |

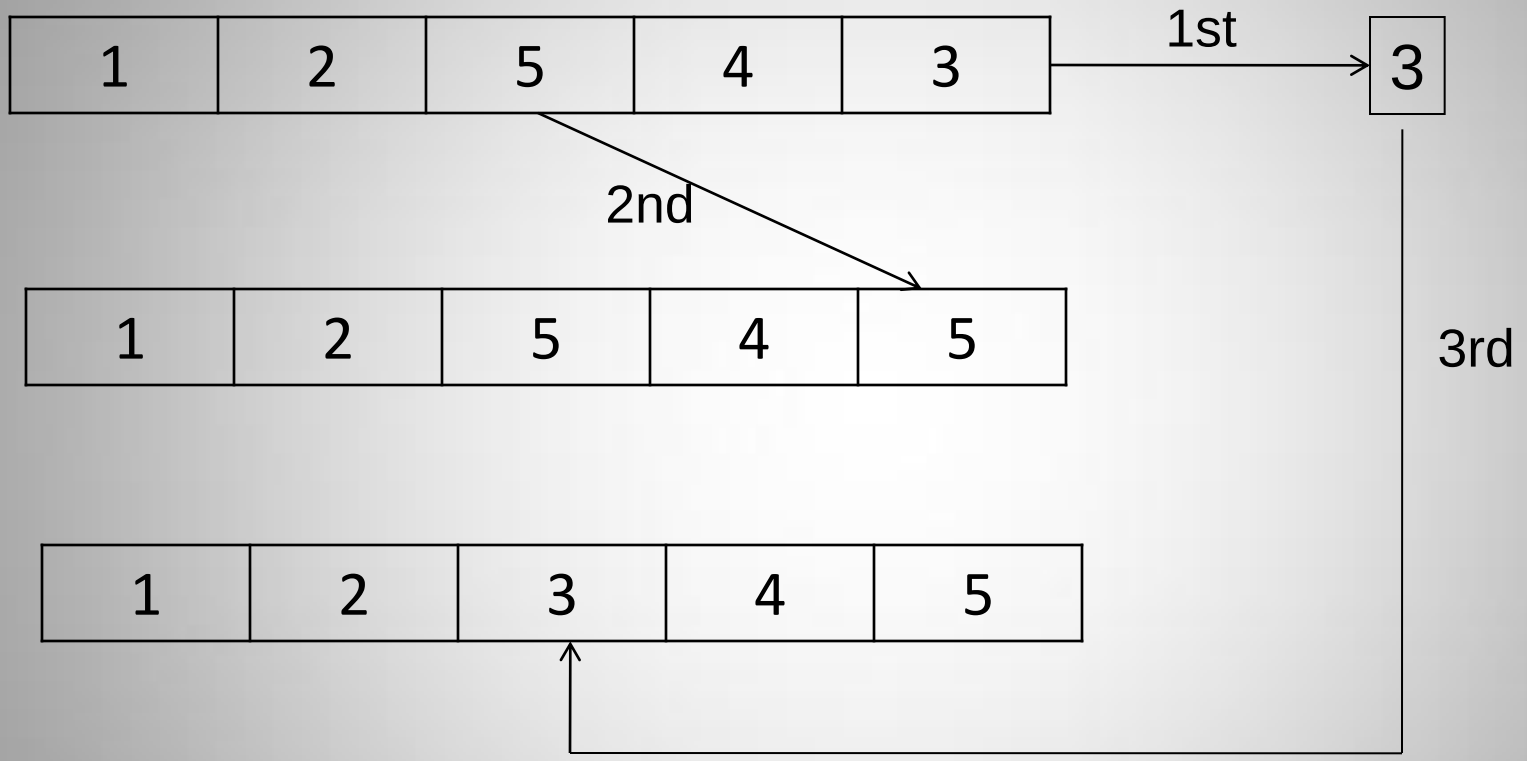
# Let's Look at Some Code

- BinSearch.cpp

# Exchanging 2 Array Elements

- You have to have an a place to hold the first element while the second element is moved into its place
- The first element can then be moved into the second element's original place

# Example



# In Code

```
int
   arr[5]={1,2,5,4,3},
   iTmp;

iTmp = arr[4];
arr[4] = arr[2];
arr[2] = iTmp;
```

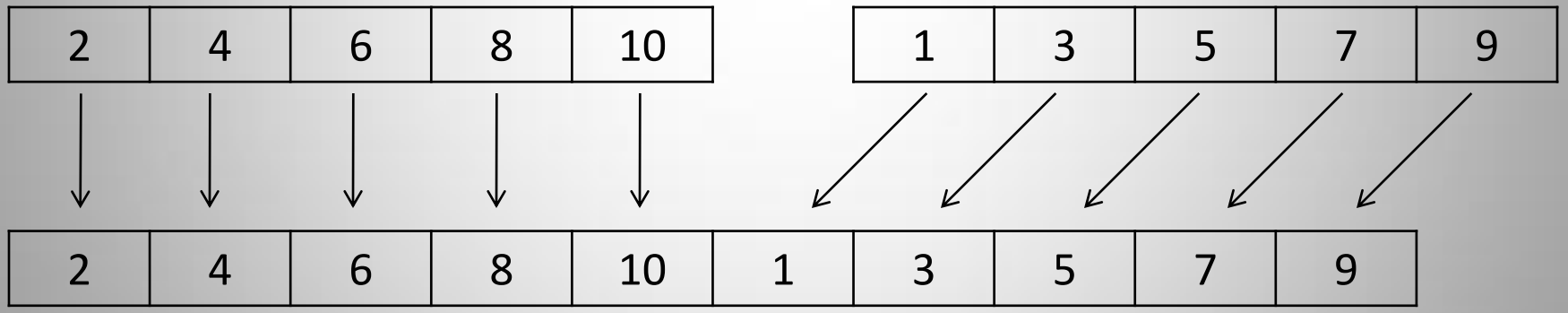
- Notice that the array subscripts arrange on the diagonaly

# In Place Manipulation

- What we just did was rearrange data in place
- We worked with the existing array

# Operations into Another Array

- Merging two arrays into one



# Code for This

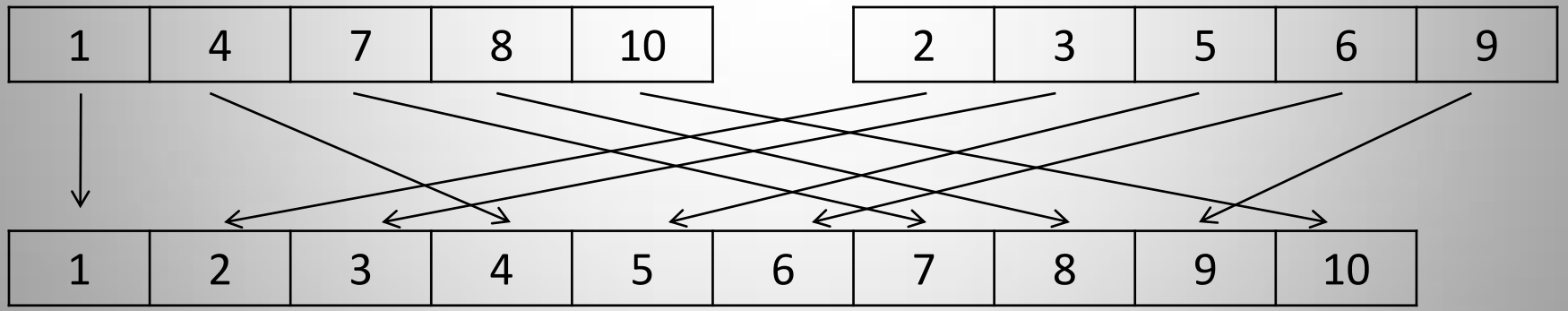
```
int
 arr1={2,4,6,8,10},
 arr2={1,3,5,7,9},
 arr3={}
 src,
 tgt = 0;
```

```
for(src=0;src<5;src++){
 arr3[tgt] = arr1[src];
 tgt++;
}

for(src=0;src<5;src++){
 arr3[tgt] = arr2[src];
 tgt++;
}
```

# Operations into Another Array

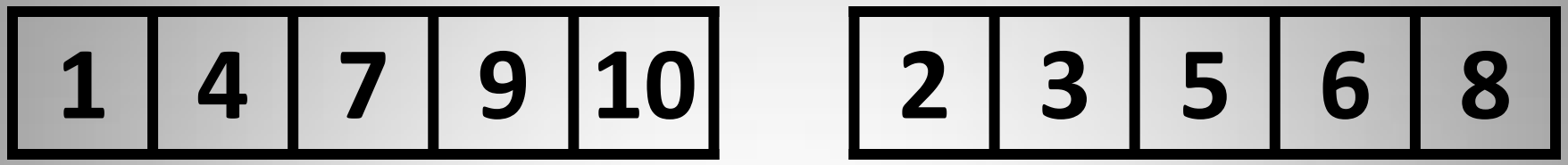
- Merging two arrays into one, maintaining sorted order



# How Do We Do That?

- How many subscript variables will we need?
- That's right, 3
- How deeply will the loops be nested?
- Gotcha – We only need one single-level loop
- For our example, what will the range of the subscripts be?
- That's right, 0 through 9
- Yes, this is confusing

# Let's Do it By hand




# How to do this in Code

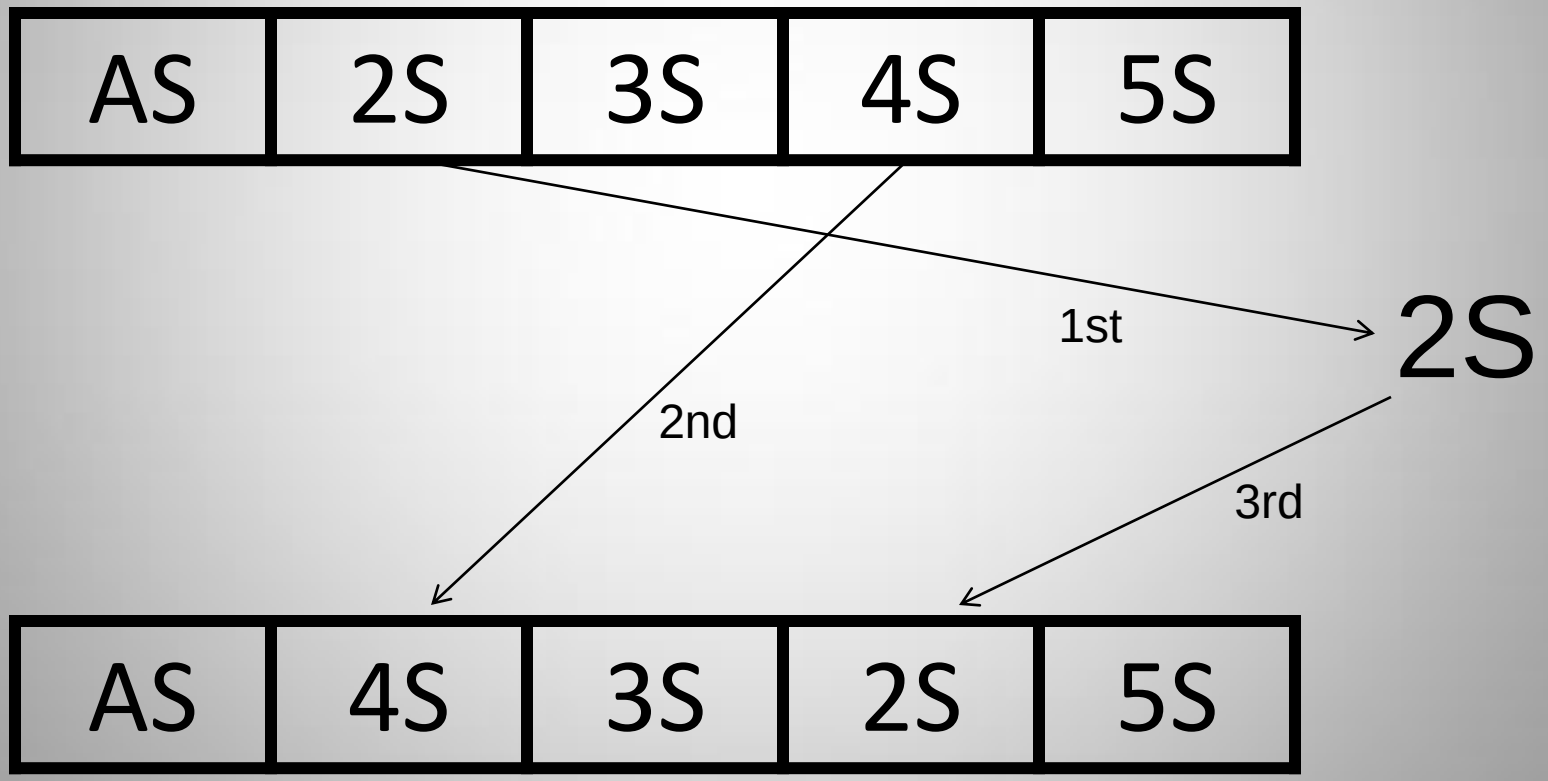
- You think that's going to fit in 2 panels on a slide?
- Think again
- Let's look at MergeSorted.cpp

# Sorting/Exchanging Objects

- You can sort based on anything you want
- All you need is a relation based on value(s) in your data
- The type of the data you sort or exchange can be anything
- Even an object

# About Lab 4

- The exchange shuffle:



# Algorithm

index 1 = random int mod 52

index 2 = random int mod 52

Store cards[index 1] in tempCard

Move cards[index 2] to card[index 1]

Move tempCard to cards[index 2]

# In C++

```
PlayingCard *tmpCard, *deck[52];
  . . .
for(i=0;i<shfls;i++){
 ndx1=rand()%52;
 while((ndx2=rand()%52) == ndx1);
 tmpCard = deck[ndx1];
 deck[ndx1] = deck[ndx2];
 deck[ndx2] = tmpCard;
}
```

# Interleave Shuffle

- How many arrays do you need?
- How many arrays are you allowed?
- How do you determine the number of cards to interleave?
- How do you make sure you don't go beyond the bounds of the halves of the deck?
- Do you count up or down?