



Chapter Six: Arrays and Vectors I

Problem Solving: Adapting Algorithms

Discovering Algorithms by Manipulating Physical Objects

Problem Solving: Adapting Algorithms

Recall that you saw quite a few
(too many?)
algorithms for working with arrays.

Suppose you need to solve a problem that
does not exactly fit any of those?

What to do?

No, “give up” is not an option!

Problem Solving: Adapting Algorithms

You can try to use algorithms you already know to produce a new algorithm that will solve this problem.

(Then you'll have yet another algorithm – even more!)

Cooking up a new algorithm!



Problem Solving: Adapting Algorithms

You can try to use algorithms you already know to produce a new algorithm that will solve this problem.

(Then you'll have yet another algorithm – even more!)

Cooking up a new algorithm!

Bon Appétit!



Problem Solving: Adapting Algorithms

Consider this problem:

Compute the final quiz score from a set of quiz scores,

but be nice:

drop the lowest score.

Problem Solving: Adapting Algorithms

Hmm, what do I know how to do?

Problem Solving: Adapting Algorithms

Calculate the sum:

```
double total = 0;
for (int i = 0; i < Size Of values; i++)
{
    total = total + values[i];
}
```

Problem Solving: Adapting Algorithms

Find the minimum:

```
double smallest = values[0];
for (int i = 1; i < SIZE Of values; i++)
{
    if (values[i] < smallest)
    {
        smallest = values[i];
    }
}
```


Problem Solving: Adapting Algorithms

Remove an element:

```
values[pos] = values[current_size - 1];  
current_size--;
```

Problem Solving: Adapting Algorithms

WOW!

I know a lot!

Problem Solving: Adapting Algorithms

Aha! Here is the algorithm:

Problem Solving: Adapting Algorithms

Aha indeed!

1. *Find the minimum*
2. *Remove it from the array*
3. *Calculate the sum*
(will be without the lowest score)
4. *Calculate the final score*



WAIT!

(Houston, we have a problem...)

Problem Solving: Adapting Algorithms

```
values[pos] = values[current_size - 1];  
current_size--;
```

This algorithm removes by knowing
the position
of the element to remove...
...but...

```
double smallest = values[0];  
for (int i = 1; i < SIZE Of values; i++)  
{  
    if (values[i] < smallest)  
    {  
        smallest = values[i];  
    }  
}
```

That's not the *position* of the smallest –
it IS the smallest.

Problem Solving: Adapting Algorithms

WOAH!

I'm not so smart after all.

Problem Solving: Adapting Algorithms

Yes!

WAIT!

(Houston, we have a another algorithm.)



Problem Solving: Adapting Algorithms

Here's another algorithm I know that *does* find the position:

```
int pos = 0;
bool found = false;
while (pos < SIZE Of values && !found)
{
    if (values[pos] == 100) // looking for 100
    {
        found = true;
    }
    else
    {
        pos++;
    }
}
```

Problem Solving: Adapting Algorithms

WOW!

Maybe I am so smart after all.

Problem Solving: Adapting Algorithms

Aha indeed!

1. *Find the minimum*
2. *Find the position of the minimum*
→ **the one I just searched for!!!**
3. *Remove it from the array*
4. *Calculate the sum*
(will be without the lowest score)
5. *Calculate the final score*



Problem Solving: Adapting Algorithms

WOAH!

Am I so smart?

I've used my algorithms to produce another.

But I'm repeating myself.

But I'm repeating myself.

Problem Solving: Adapting Algorithms

THIS IS NOT GOOD!
THIS IS NOT GOOD!

Problem Solving: Adapting Algorithms

I searched
and then
I searched.

Problem Solving: Adapting Algorithms

I searched
for the minimum
and then
I searched
for the position...
...of the minimum!

Problem Solving: Adapting Algorithms

I'm repeating myself.
I'm repeating myself.

WAIT!

(Houston, we have a...

WAIT!

(Houston, we have a... thought...)

Problem Solving: Adapting Algorithms

I wonder if I can *adapt* the algorithm
that finds the minimum so that it finds
the position of the minimum?

Problem Solving: Adapting Algorithms

WOW!

I really am smart!

(if I can pull this off)

Problem Solving: Adapting Algorithms

I'll start with this:

```
double smallest = values[0];  
for (int i = 1; i < Size Of values; i++)  
{  
    if (values[i] < smallest)  
    {  
        smallest = values[i];  
    }  
}
```

Problem Solving: Adapting Algorithms

What is it about the minimum value
and where the minimum value is?

```
double smallest = values[0];  
for (int i = 1; i < SIZE Of values; i++)  
{  
    if (values[i] < smallest)  
    {  
        smallest = values[i];  
    }  
}
```

Problem Solving: Adapting Algorithms

What is it about the minimum value
and where the minimum value is?

```
int smallest_position = 0;
for (int i = 1; i < Size Of values; i++)
{
    if (values[i] < values[smallest_position])
    {
        smallest_position = i;
    }
}
```

Problem Solving: Adapting Algorithms

What is it about the minimum value
and where the minimum value is?

```
double smallest = values[0];  
for (int i = 1; i < SIZE Of values; i++)  
{  
    if (values[i] < smallest)  
    {  
        smallest = values[i];  
    }  
}
```


Problem Solving: Adapting Algorithms

What is it about the minimum value
and where the minimum value is?

```
int smallest_position = 0;
for (int i = 1; i < Size Of values; i++)
{
    if (values[i] < values[smallest_position])
    {
        smallest_position = i;
    }
}
```

Problem Solving: Adapting Algorithms

What is it about the minimum value
and where the minimum value is?

```
double smallest = values[0];  
for (int i = 1; i < SIZE Of values; i++)  
{  
    if (values[i] < smallest)  
    {  
        smallest = values[i];  
    }  
}
```

Problem Solving: Adapting Algorithms

What is it about the minimum value
and where the minimum value is?

```
int smallest_position = 0;
for (int i = 1; i < Size Of values; i++)
{
    if (values[i] < values[smallest_position])
    {
        smallest_position = i;
    }
}
```

Problem Solving: Adapting Algorithms

There it is!

```
int smallest_position = 0;
for (int i = 1; i < Size Of values; i++)
{
    if (values[i] < values[smallest_position])
    {
        smallest_position = i;
    }
}
```

Problem Solving: Adapting Algorithms

Aha.

Problem Solving: Adapting Algorithms

Aha indeed!

1. Find the **position** of the minimum
2. Remove it from the array
3. Calculate the sum
(will be without the lowest score)
4. Calculate the final score



Problem Solving: Adapting Algorithms

Now *I'm*
repeating
myself!

1. **Position** of the minimum
2. Remove it from the array
3. Calculate the sum
(will be without the lowest score)
4. Calculate the final score



Discovering Algorithms by Manipulating Physical Objects

What if you come across a problem
for which you cannot find an algorithm you know
and you cannot figure out how to adapt any algorithms?

What to do?

No, again, “give up” is not an option!

There is a technique that you can use called:

MANIPULATING PHYSICAL OBJECTS

better know as:

playing around with things.

Discovering Algorithms by Manipulating Physical Objects



Playin roun!

Discovering Algorithms by Manipulating Physical Objects

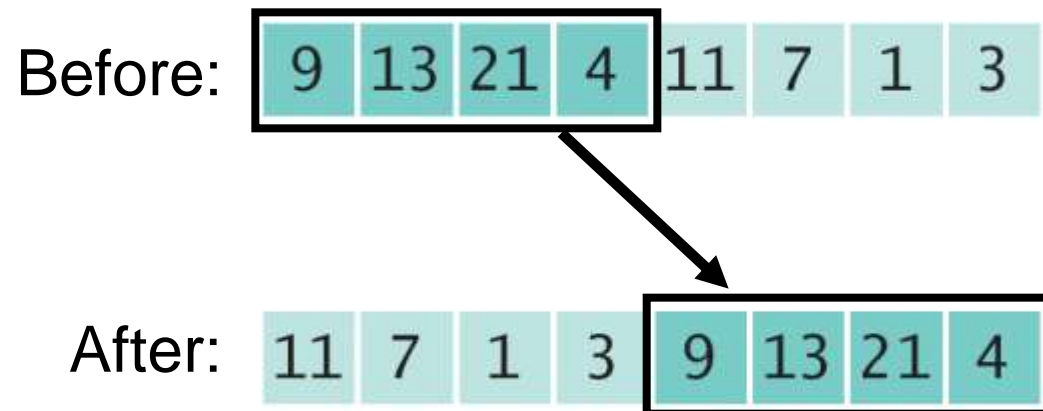


Playin roun! Wiff tings!

Discovering Algorithms by Manipulating Physical Objects

Here is a problem:

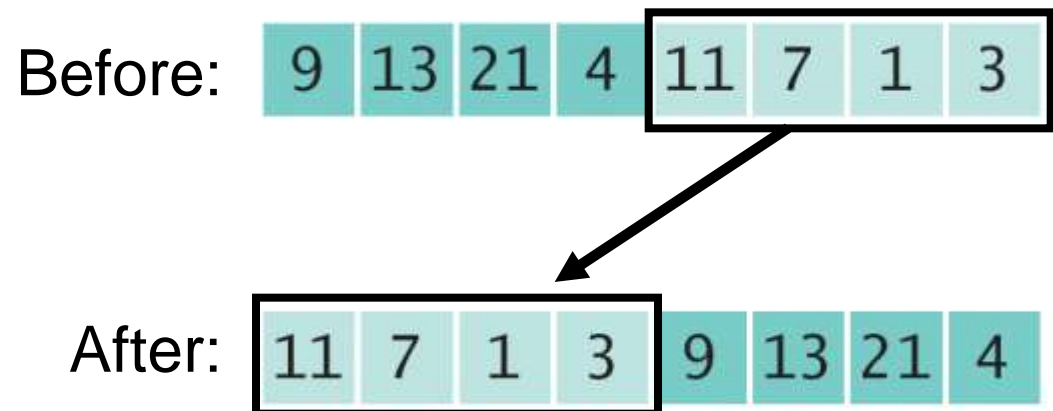
You are given an array whose size is an even number.
You are to switch the first and the second half.



Discovering Algorithms by Manipulating Physical Objects

Here is a problem:

You are given an array whose size is an even number.
You are to switch the first and the second half.



Discovering Algorithms by Manipulating Physical Objects



Less star playin!

Discovering Algorithms by Manipulating Physical Objects

To learn this *Manipulating Physical Objects* technique,
let's play with some coins
and review some algorithms you already know.

OK, let's *manipulate* some coins.
Go get eight coins.

Discovering Algorithms by Manipulating Physical Objects

Thanks Penny,
but we need an assortment of different coins.



Discovering Algorithms by Manipulating Physical Objects

Good.



Discovering Algorithms by Manipulating Physical Objects

What algorithms do you know
that allow you to rearrange a set of coins?



Discovering Algorithms by Manipulating Physical Objects

You know how to remove a coin.



You! Be gone!

Discovering Algorithms by Manipulating Physical Objects

You know how to remove a coin.



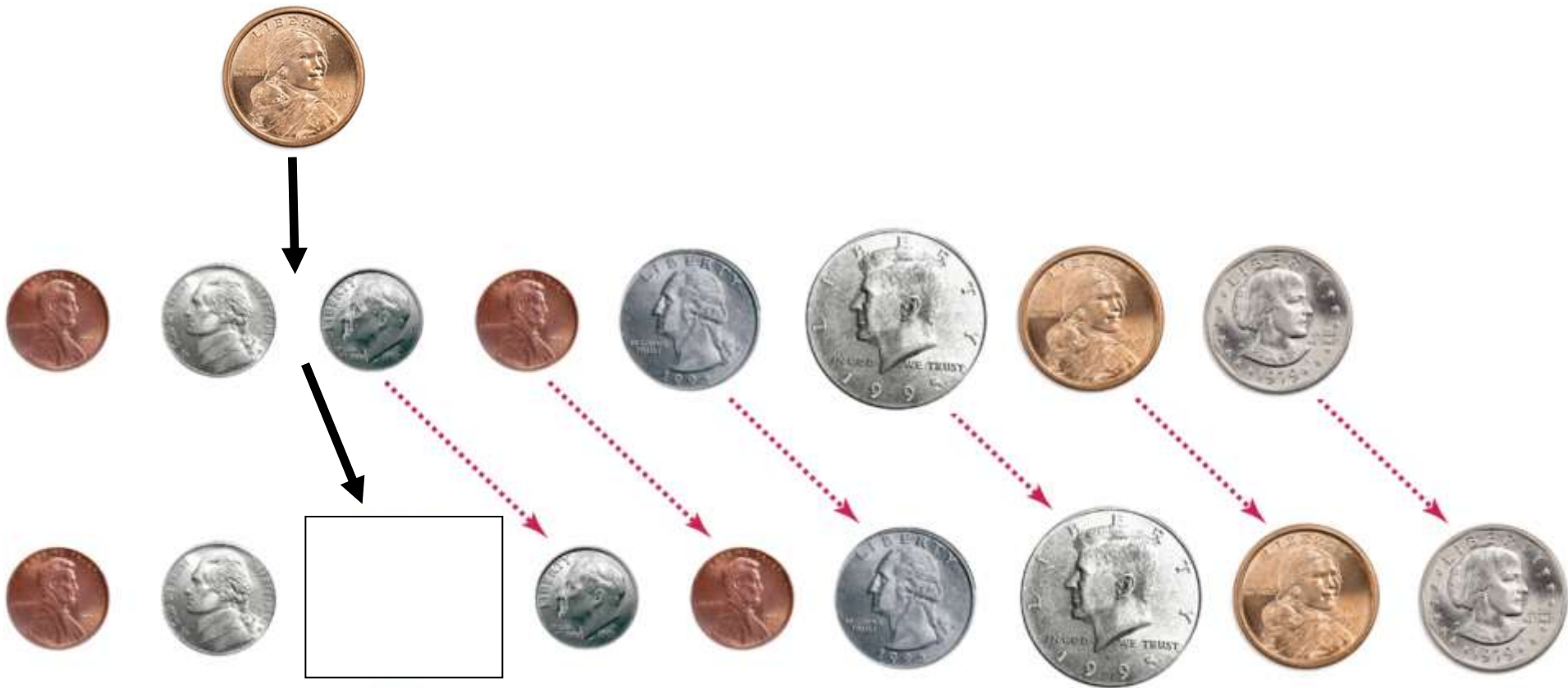
Discovering Algorithms by Manipulating Physical Objects

You know how to insert a coin at a specific position.



Discovering Algorithms by Manipulating Physical Objects

You know how to insert a coin at a specific position.



Discovering Algorithms by Manipulating Physical Objects

You know how to insert a coin at a specific position.



Discovering Algorithms by Manipulating Physical Objects

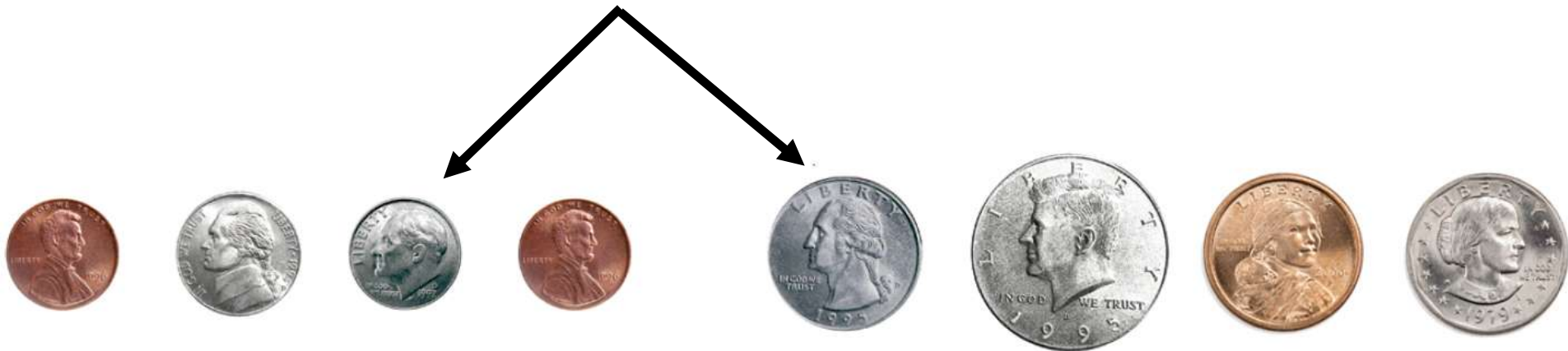


Playin wiff algribbms sure ib fun!

Discovering Algorithms by Manipulating Physical Objects

And you know how to swap two elements.

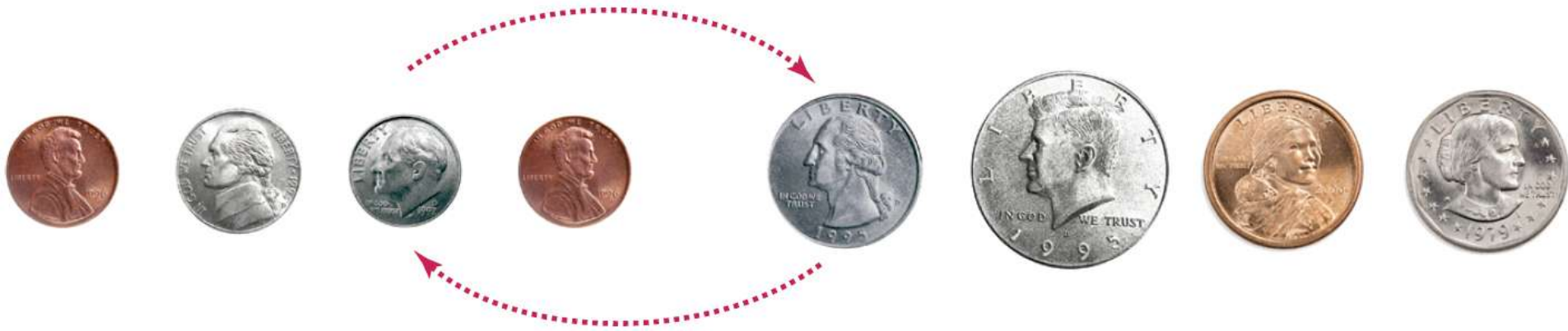
You two!



Swap places!

Discovering Algorithms by Manipulating Physical Objects

And you know how to swap two elements.



Discovering Algorithms by Manipulating Physical Objects

And you know how to swap two elements.



Discovering Algorithms by Manipulating Physical Objects



What you mean? Gih 2 work?

Discovering Algorithms by Manipulating Physical Objects



What you mean? Star stinkin?

Discovering Algorithms by Manipulating Physical Objects

Get to work.

Start THINK-ing.

Discovering Algorithms by Manipulating Physical Objects

Swapping.



Discovering Algorithms by Manipulating Physical Objects

Swapping any two.

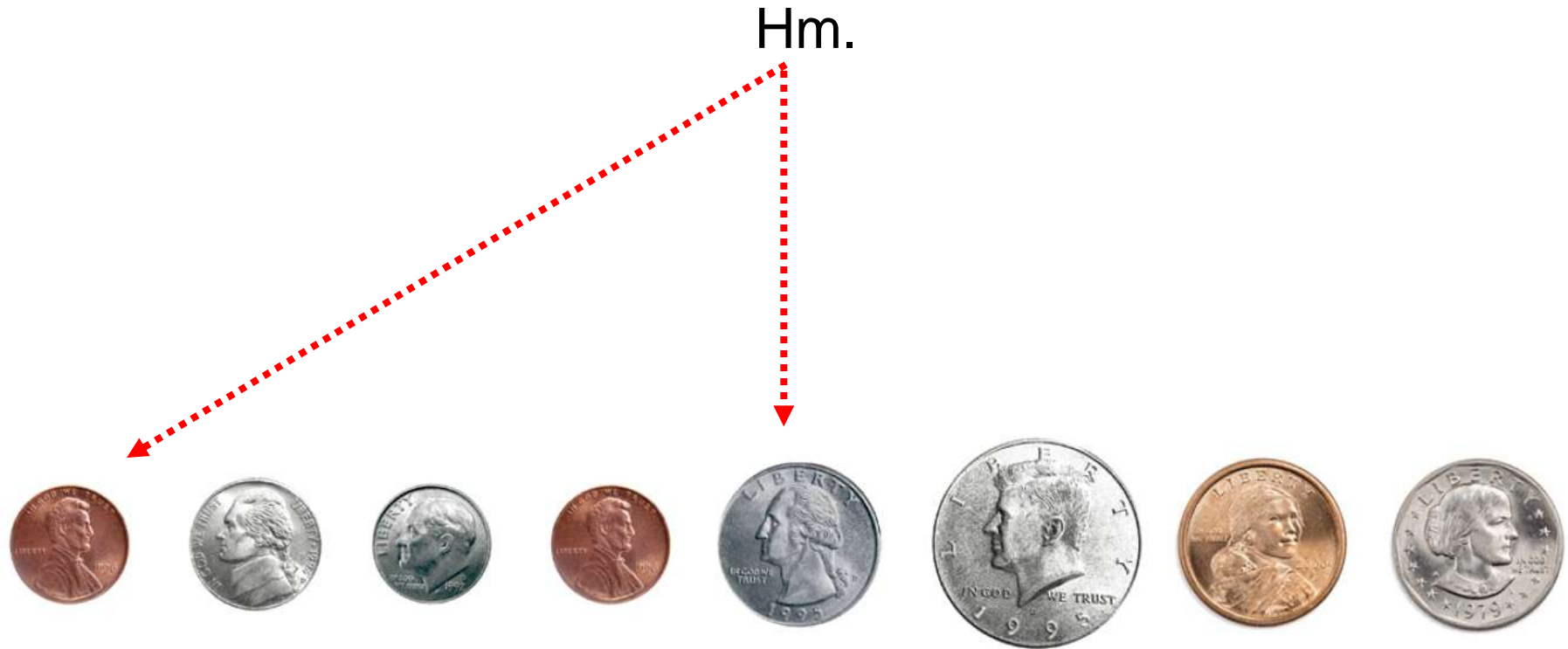


Discovering Algorithms by Manipulating Physical Objects

Any two.

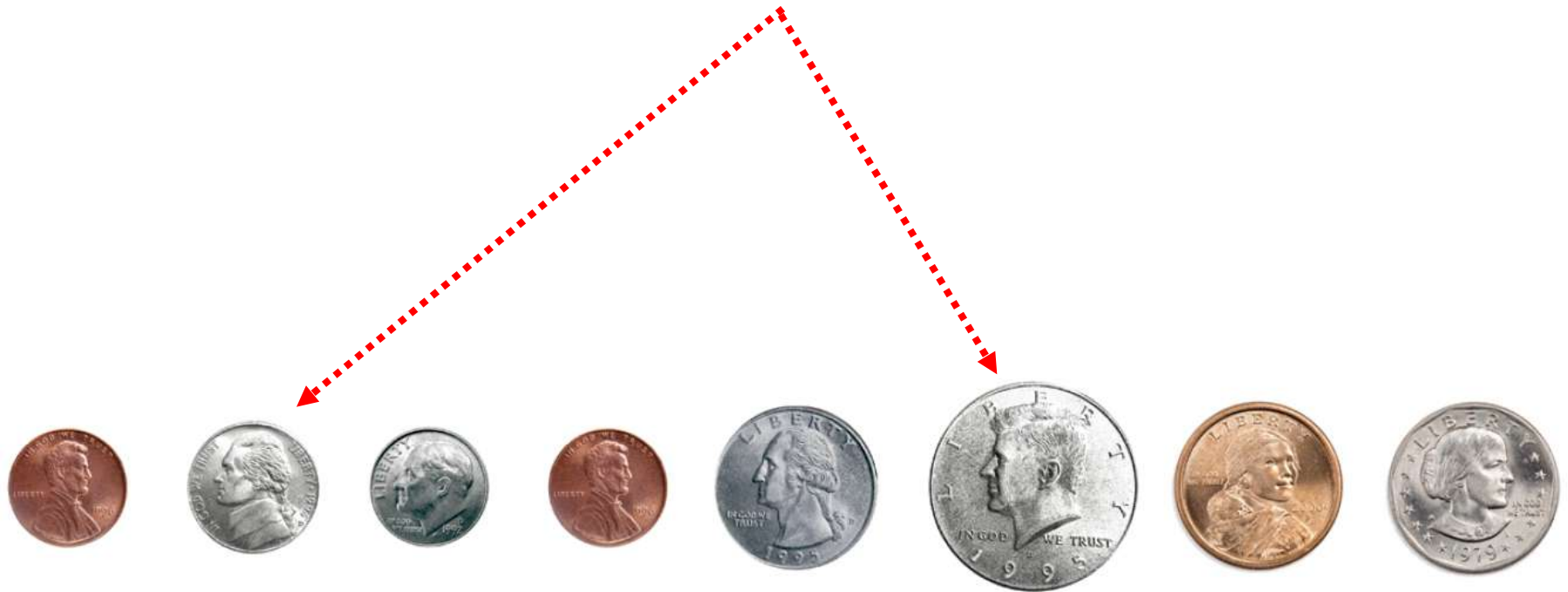


Discovering Algorithms by Manipulating Physical Objects



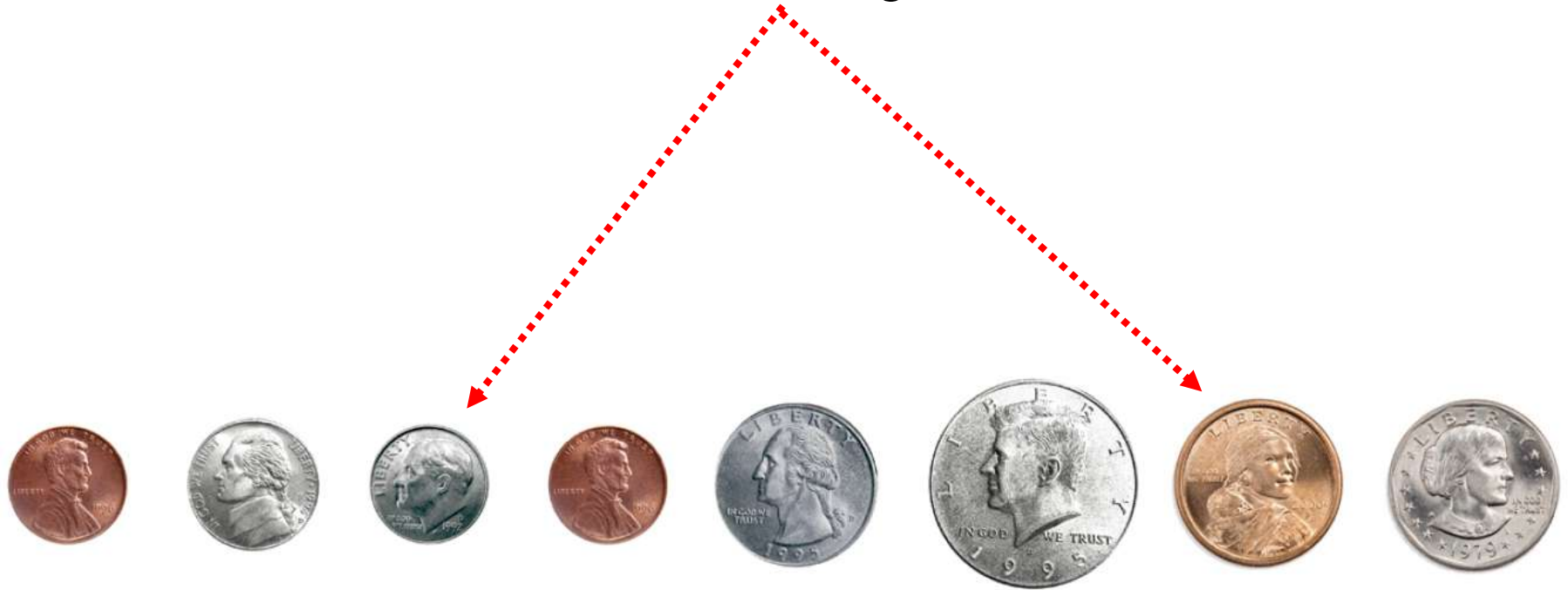
Discovering Algorithms by Manipulating Physical Objects

And hm.



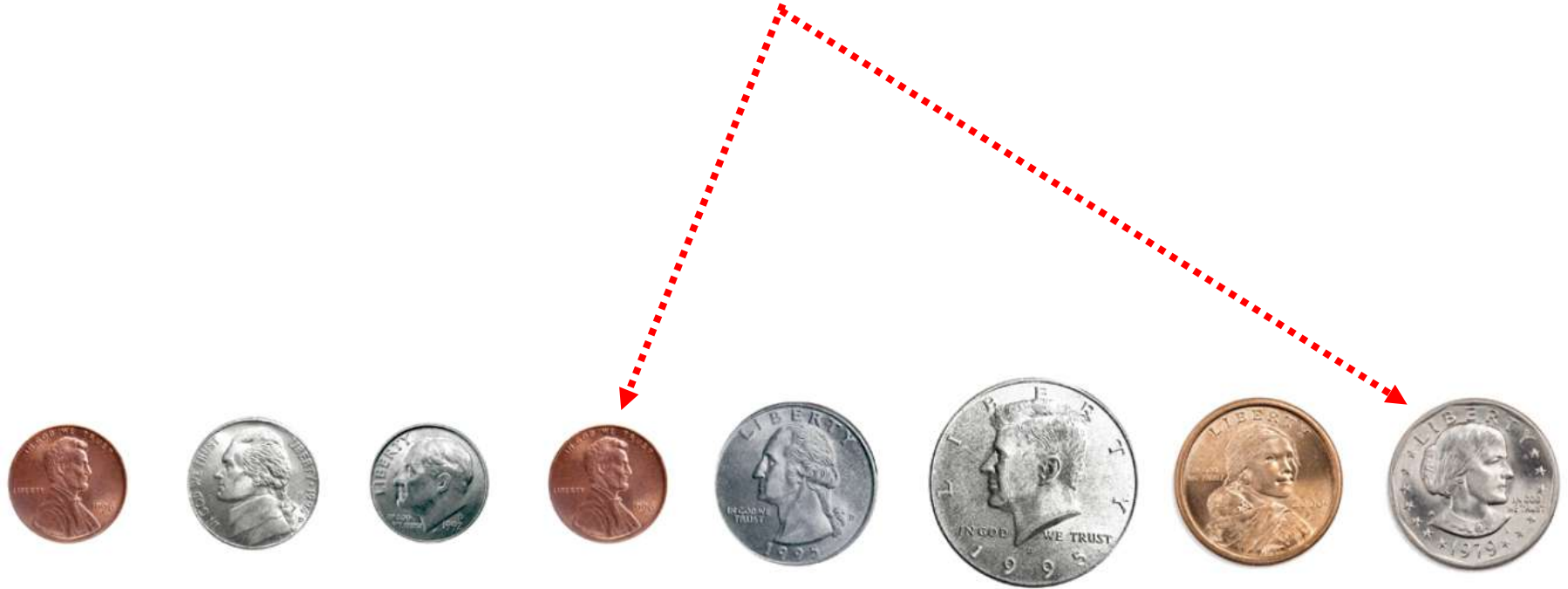
Discovering Algorithms by Manipulating Physical Objects

Then hm again.

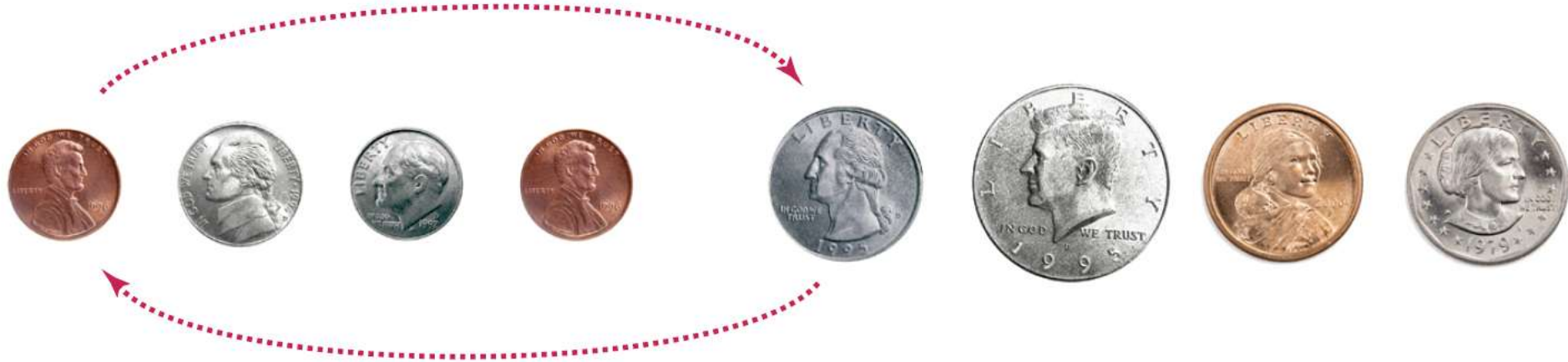


Discovering Algorithms by Manipulating Physical Objects

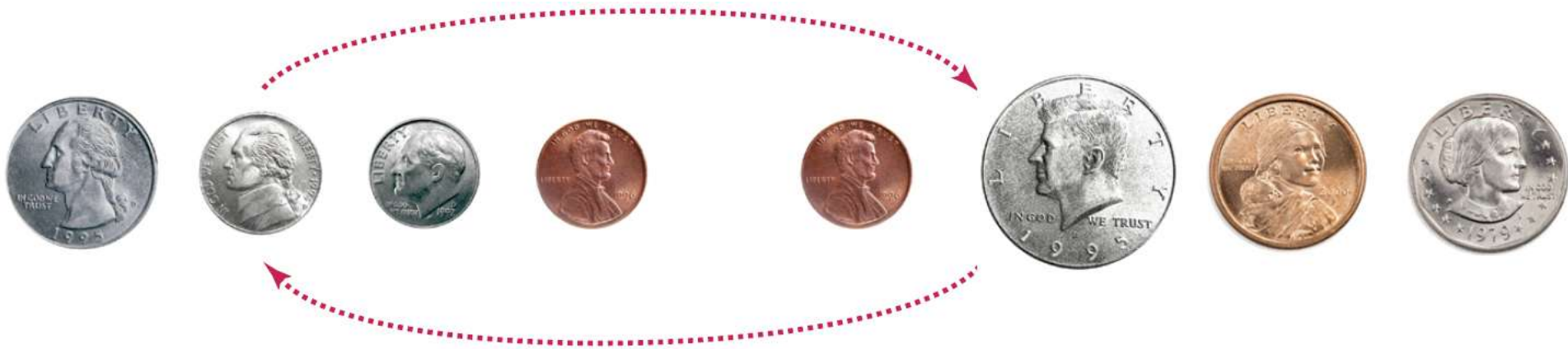
And finally...



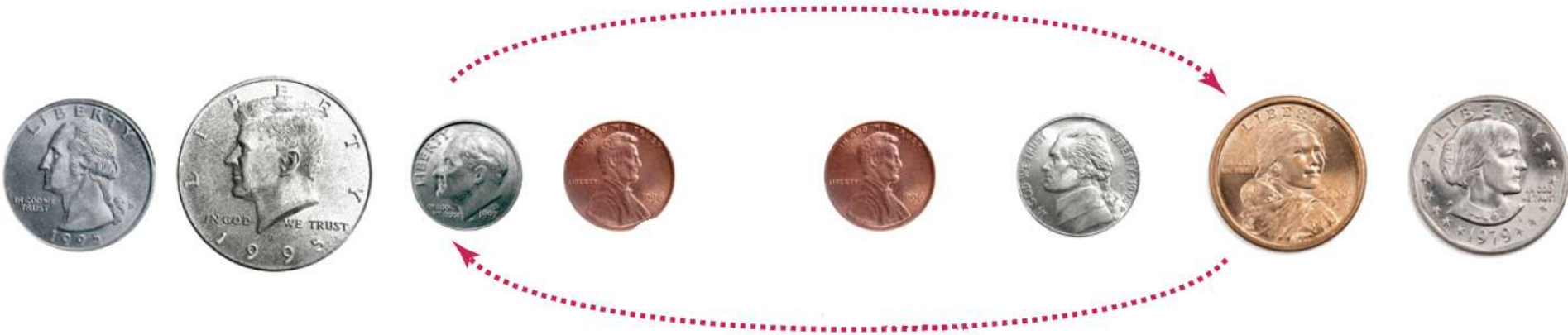
Discovering Algorithms by Manipulating Physical Objects



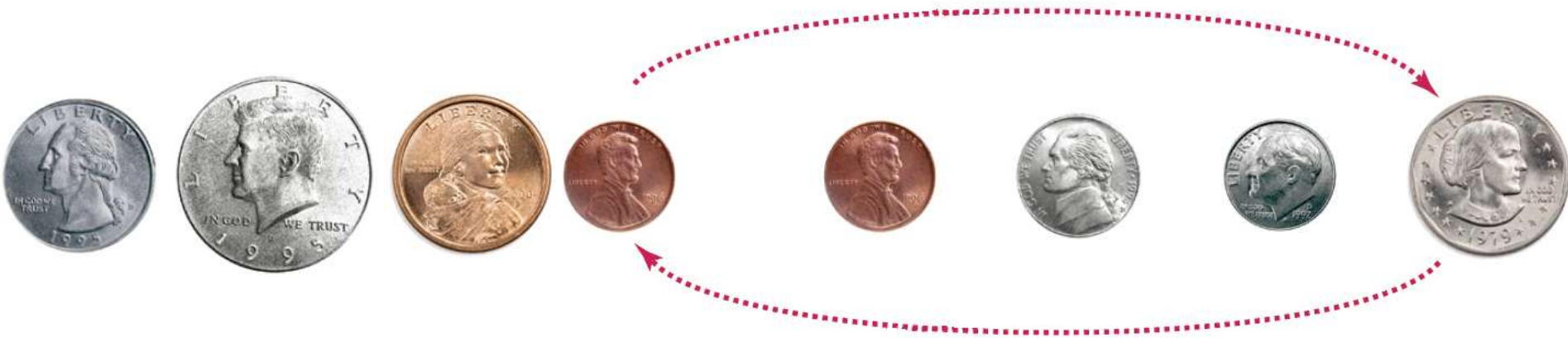
Discovering Algorithms by Manipulating Physical Objects



Discovering Algorithms by Manipulating Physical Objects



Discovering Algorithms by Manipulating Physical Objects



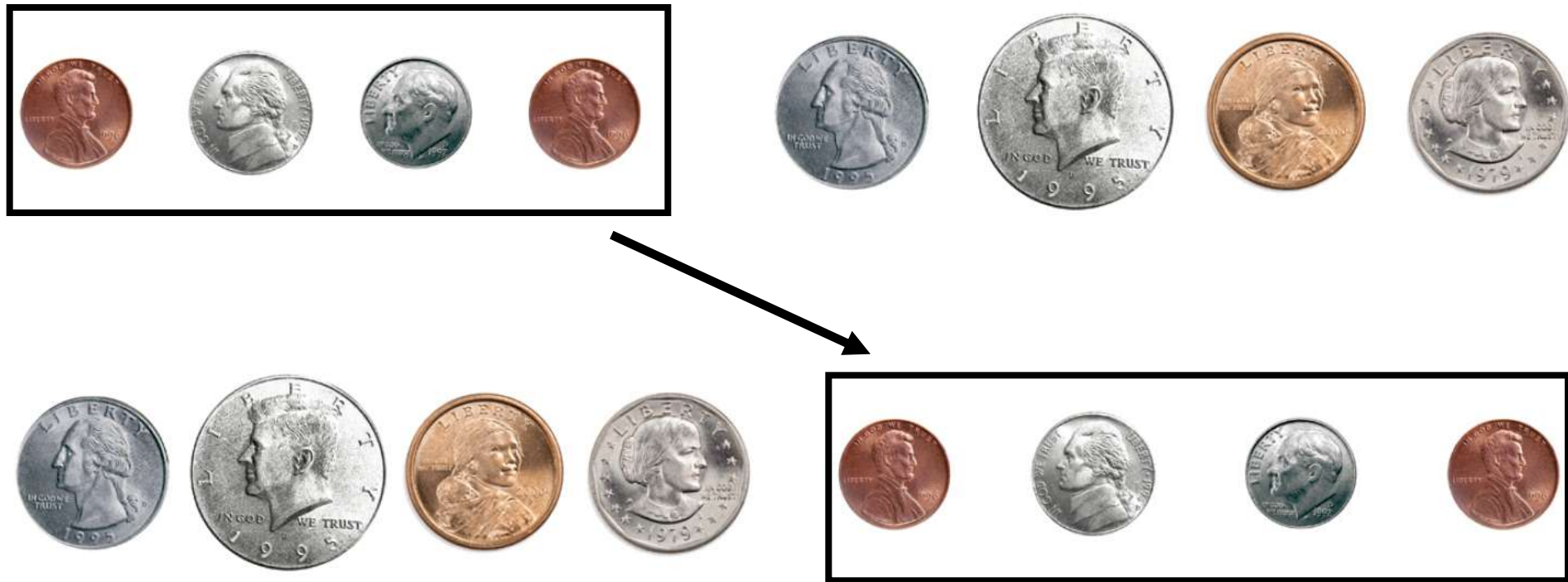
Discovering Algorithms by Manipulating Physical Objects



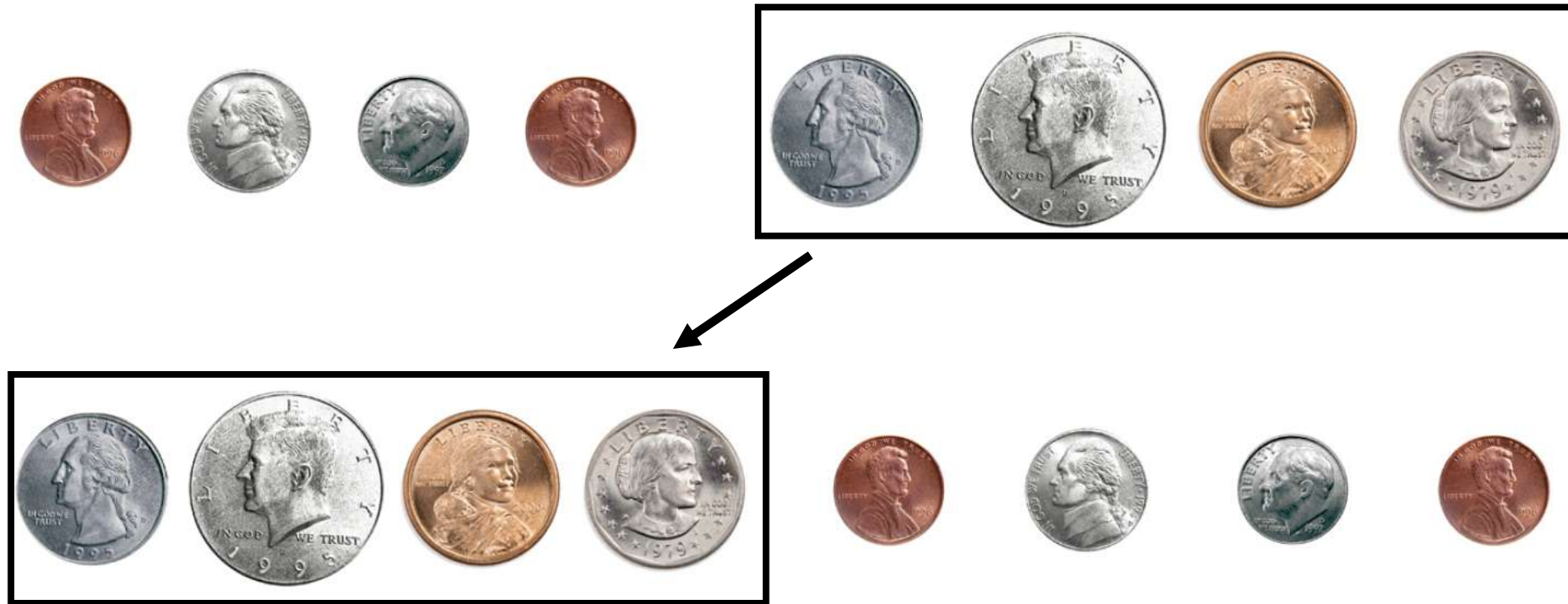
AHA!



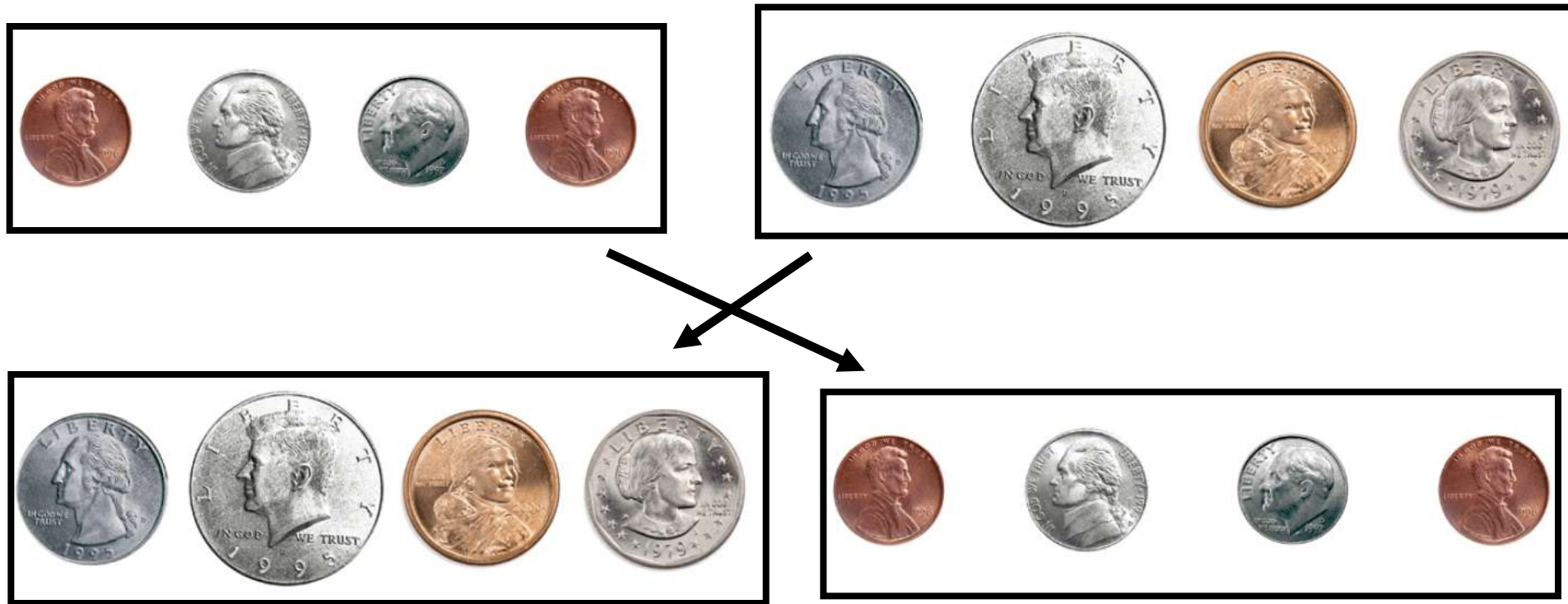
Discovering Algorithms by Manipulating Physical Objects



Discovering Algorithms by Manipulating Physical Objects



Discovering Algorithms by Manipulating Physical Objects



Discovering Algorithms by Manipulating Physical Objects

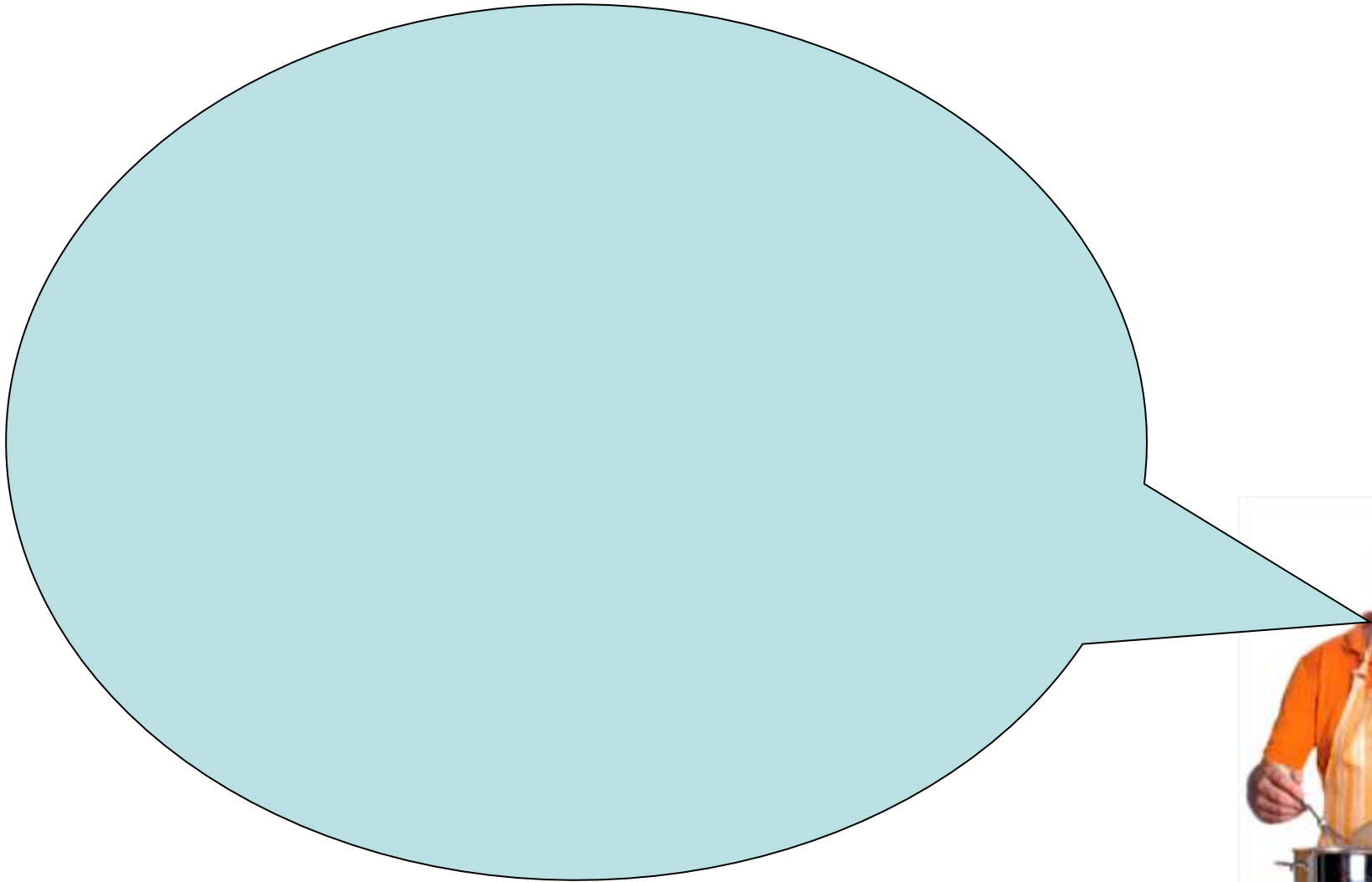
AHA!
INDEED!

Now for the
algorithm
– in code.

I'm going to
need a bigger
balloon!



Discovering Algorithms by Manipulating Physical Objects



Discovering Algorithms by Manipulating Physical Objects

`i =`
`j =`

Two indices means we
need two variables.



Discovering Algorithms by Manipulating Physical Objects

`i =`
`j =`

Initialization?



Discovering Algorithms by Manipulating Physical Objects

```
i = 0;  
j =
```

OK.



Discovering Algorithms by Manipulating Physical Objects

```
i = 0;  
j =
```



Discovering Algorithms by Manipulating Physical Objects

```
i = 0;  
j = ?
```

Where does that
index start?



Discovering Algorithms by Manipulating Physical Objects

```
i = 0;  
j = ?
```

We swap the leftmost
with somewhere in
the middle.



Discovering Algorithms by Manipulating Physical Objects

```
i = 0;  
j = ?
```

The middle!
That's it – half way
into the array.



Discovering Algorithms by Manipulating Physical Objects

```
i = 0;  
j = size / 2;
```

Now we will loop...



Discovering Algorithms by Manipulating Physical Objects

```
i = 0;  
j = size / 2;  
while ( ??? )
```

...but...
for how long?
until when?



Discovering Algorithms by Manipulating Physical Objects

```
i = 0;  
j = size / 2;  
while ( ??? )
```

Let's think
about that later.



Discovering Algorithms by Manipulating Physical Objects

```
i = 0;  
j = size / 2;  
while ( ??? )
```

For certain we will
be swapping the
elements at the
indices...



Discovering Algorithms by Manipulating Physical Objects

```
i = 0;  
j = size / 2;  
while ( ??? )  
    swap elements at i and j
```

...and then go on to
the next pair of
indices to swap...



Discovering Algorithms by Manipulating Physical Objects

```
i = 0;  
j = size / 2;  
while ( ??? )  
    swap elements at i and j  
    i++;  
    j++;
```

But when are we
finished swapping?



Discovering Algorithms by Manipulating Physical Objects

```
i = 0;  
j = size / 2;  
while ( ??? )  
    swap elements at i and j  
    i++;  
    j++;
```

We only process
half the array



Discovering Algorithms by Manipulating Physical Objects

```
i = 0;  
j = size / 2;  
while ( ??? )  
    swap elements at i and j  
    i++;  
    j++;
```

AHA!



Discovering Algorithms by Manipulating Physical Objects

```
i = 0;  
j = size / 2;  
while (i < size / 2)  
    swap elements at i and j  
    i++;  
    j++;
```

That's the algorithm!



Discovering Algorithms by Manipulating Physical Objects



Can I stop stinkin?

Discovering Algorithms by Manipulating Physical Objects



***Can I stop stinkin?
an gih bat to playin?***



End Arrays and Vectors I

Problem Solving: Adapting Algorithms

Discovering Algorithms by Manipulating Physical Objects