10.6.1
Proving that merge is correct

Proving Correctness

Obvious way to prove correctness of recursive algorithm: induction!

- Easy to show by induction on n that MergeSort is correct if you assume Merge is correct.
- How do we prove that Merge is correct? Also by induction!
- One way is to rewrite Merge into a recursive version.
- For algorithms with loops one comes up with a natural loop invariant that captures all the essential properties and then we prove the loop invariant by induction on the index of the loop.

Proving Correctness

Obvious way to prove correctness of recursive algorithm: induction!

- Easy to show by induction on n that MergeSort is correct if you assume Merge is correct.
- How do we prove that Merge is correct? Also by induction!
- One way is to rewrite Merge into a recursive version.
- For algorithms with loops one comes up with a natural loop invariant that captures all the essential properties and then we prove the loop invariant by induction on the index of the loop.

Merge is correct..

```
Merge(A[1...m], A[m + 1...n])
    i ← 1, j ← m + 1, k ← 1
    while ( k ≤ n ) do
        if i > m or (j ≤ n and A[i] > A[j])
            B[k++] ← A[j++]
        else
            B[k++] ← A[i++]
    A ← B
```

Claim

Assuming $A[1...m]$ and $A[m+1...n]$ are sorted (all values distinct). For any value of k, in the beginning of the loop, we have:

1. $B[1...k-1]$ *contains the* $k-1$ *smallest elements in* A.
2. $B[1...k-1]$ *is sorted.*

Merge is correct..

```
Merge(A[1...m], A[m+1...n])
    i ← 1, j ← m+1, k ← 1
    while ( k ≤ n ) do
        if i > m or (j ≤ n and A[i] > A[j])
            B[k++] ← A[j++]
        else
            B[k++] ← A[i++]
    A ← B
```

Claim

Assuming $A[1...m]$ *and* $A[m+1...n]$ *are sorted (all values distinct).*
For any value of k, *in the beginning of the loop, we have:*

1. $B[1...k-1]$ *contains the* $k-1$ *smallest elements in* A.
2. $B[1...k-1]$ *is sorted.*

Merge is correct

```
Merge(A[1...m], A[m+1...n])
i ← 1, j ← m+1, k ← 1
while ( k ≤ n ) do
    if i > m or (j ≤ n and A[i] > A[j])
        B[k++] ← A[j++]
    else
        B[k++] ← A[i++]
A ← B
```

Claim

Assuming $A[1...m]$ *and* $A[m+1...n]$ *are sorted (all values distinct).*
$\forall k$, *in beginning of the loop, we have:*

1. $B[1...k-1]$: $k-1$ *smallest elements in* A.
2. $B[1...k-1]$ *is sorted.*

Proof:

Merge is correct

```
Merge(A[1...m], A[m + 1...n])
i ← 1, j ← m + 1, k ← 1
while ( k ≤ n ) do
    if i > m or (j ≤ n and A[i] > A[j])
        B[k + +] ← A[j + +]
    else
        B[k + +] ← A[i + +]
A ← B
```

Claim

Assuming $A[1 \ldots m]$ and $A[m+1 \ldots n]$ are sorted (all values distinct).
$\forall k$, in beginning of the loop, we have:

1. $B[1 \ldots k-1]$: *$k-1$ smallest elements in A.*
2. $B[1 \ldots k-1]$ *is sorted.*

Proof:

Base of induction: $k = 1$: Emptily true.

Merge is correct

```
Merge(A[1...m], A[m + 1...n])
i ← 1, j ← m + 1, k ← 1
while ( k ≤ n ) do
    if i > m or (j ≤ n and A[i] > A[j])
        B[k++] ← A[j++]
    else
        B[k++] ← A[i++]
A ← B
```

Claim

Assuming $A[1...m]$ *and* $A[m+1...n]$ *are sorted (all values distinct).*
$\forall k$, *in beginning of the loop, we have:*

1. $B[1...k-1]$: $k-1$ *smallest elements in* A.
2. $B[1...k-1]$ *is sorted.*

Proof:

Inductive hypothesis: Claim true for all $k \leq \alpha$.

Merge is correct

```
Merge(A[1...m], A[m+1...n])
i ← 1, j ← m+1, k ← 1
while ( k ≤ n ) do
    if i > m or (j ≤ n and A[i] > A[j])
        B[k++] ← A[j++]
    else
        B[k++] ← A[i++]
A ← B
```

Claim

Assuming $A[1 \ldots m]$ *and* $A[m+1 \ldots n]$ *are sorted (all values distinct).*
$\forall k$, *in beginning of the loop, we have:*

1. $B[1 \ldots k-1]$: $k-1$ *smallest elements in* A.
2. $B[1 \ldots k-1]$ *is sorted.*

Proof:

Inductive hypothesis: Claim true for all $k \leq \alpha$.
Inductive step: Need to prove claim true for $k = \alpha + 1$.

Merge is correct

```
Merge(A[1...m], A[m + 1...n])
i ← 1, j ← m + 1, k ← 1
while ( k ≤ n ) do
    if i > m or (j ≤ n and A[i] > A[j])
        B[k + +] ← A[j + +]
    else
        B[k + +] ← A[i + +]
A ← B
```

Claim

Assuming $A[1...m]$ and $A[m+1...n]$ are sorted (all values distinct).
$\forall k$, in beginning of the loop, we have:

1. *$B[1...k-1]$: $k-1$ smallest elements in A.*
2. *$B[1...k-1]$ is sorted.*

Inductive hypothesis: Claim true for all $k \leq \alpha$.

Idea: Start at iteration $k = \alpha$, and use induction hypothesis, run the loop for one iter...

Merge is correct

```
Merge(A[1...m], A[m + 1...n])
i ← 1, j ← m + 1, k ← 1
while ( k ≤ n ) do
    if i > m or (j ≤ n and A[i] > A[j])
        B[k + +] ← A[j + +]
    else
        B[k + +] ← A[i + +]
A ← B
```

Claim

Assuming $A[1...m]$ and $A[m+1...n]$ are sorted (all values distinct).
$\forall k$, in beginning of the loop, we have:

1. *$B[1...k-1]$: $k-1$ smallest elements in A.*
2. *$B[1...k-1]$ is sorted.*

Inductive hypothesis: Claim true for all $k \leq \alpha$.
Idea: Start at iteration $k = \alpha$, and use induction hypothesis, run the loop for one iter...
If $i > m$ then true.

Merge is correct

```
Merge(A[1...m], A[m+1...n])
i ← 1, j ← m+1, k ← 1
while ( k ≤ n ) do
    if i > m or (j ≤ n and A[i] > A[j])
        B[k++] ← A[j++]
    else
        B[k++] ← A[i++]
A ← B
```

Claim

Assuming $A[1\ldots m]$ and $A[m+1\ldots n]$ are sorted (all values distinct).
$\forall k$, in beginning of the loop, we have:

1. *$B[1\ldots k-1]$: $k-1$ smallest elements in A.*
2. *$B[1\ldots k-1]$ is sorted.*

Inductive hypothesis: Claim true for all $k \leq \alpha$.

Idea: Start at iteration $k = \alpha$, and use induction hypothesis, run the loop for one iter...

If $i > m$ then true.

If $j > n$ then true.

Merge is correct

```
Merge(A[1...m], A[m+1...n])
i ← 1, j ← m+1, k ← 1
while ( k ≤ n ) do
    if i > m or (j ≤ n and A[i] > A[j])
        B[k++] ← A[j++]
    else
        B[k++] ← A[i++]
A ← B
```

Claim

Assuming $A[1...m]$ and $A[m+1...n]$ are sorted (all values distinct).
$\forall k$, in beginning of the loop, we have:

1. $B[1...k-1]$: $k-1$ *smallest elements in* A.
2. $B[1...k-1]$ *is sorted.*

Inductive hypothesis: Claim true for all $k \leq \alpha$.
Idea: Start at iteration $k = \alpha$, and use induction hypothesis, run the loop for one iter...
If $i \leq m$ and $j \leq n$ then...

Merge is correct

```
Merge(A[1...m], A[m + 1...n])
i ← 1, j ← m + 1, k ← 1
while ( k ≤ n ) do
    if i > m or (j ≤ n and A[i] > A[j])
        B[k + +] ← A[j + +]
    else
        B[k + +] ← A[i + +]
A ← B
```

Claim

Assuming $A[1 \ldots m]$ and $A[m+1 \ldots n]$ are sorted (all values distinct).
$\forall k$, in beginning of the loop, we have:

1. *$B[1 \ldots k-1]$: $k-1$ smallest elements in A.*
2. *$B[1 \ldots k-1]$ is sorted.*

Inductive hypothesis: Claim true for all $k \leq \alpha$.
Idea: Start at iteration $k = \alpha$, and use induction hypothesis, run the loop for one iter...
If $i \leq m$ and $j \leq n$ then...

□

Merge is correct!!!

Claim

Assuming $A[1 \ldots m]$ and $A[m+1 \ldots n]$ are sorted (all values distinct).
$\forall k$, in beginning of the loop, we have:

1. $B[1 \ldots k-1]$: $k-1$ *smallest elements in* A.
2. $B[1 \ldots k-1]$ *is sorted.*

Proved claim is correct. Plugging $k = n+1$, implies.

Claim

By end of loop execution B *(and thus* A*) contain the elements of* A *in sorted order.*
$\implies$ **Merge** *is correct.*

THE END

...

(for now)