10.6.2
Proving that merge-sort is correct

Proving correctness of merge-sort

```
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
```

```
MergeSort(A[1...n])
    if n ≤ 1 then return
    m ← ⌊n/2⌋
    MergeSort(A[1...m])
    MergeSort(A[m + 1...n])
    Merge(A[1...m], A(m + 1...n])
```

Proved: Merge is correct.

Proving correctness of merge-sort

```
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
```

```
MergeSort(A[1...n])
    if n ≤ 1 then return
    m ← ⌊n/2⌋
    MergeSort(A[1...m])
    MergeSort(A[m+1...n])
    Merge(A[1...m], A(m+1...n])
```

Proved: Merge is correct.

Lemma

MergeSort *correctly sort the input array.*

Proving correctness of merge-sort

```
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
```

```
MergeSort(A[1...n])
    if n ≤ 1 then return
    m ← ⌊n/2⌋
    MergeSort(A[1...m])
    MergeSort(A[m+1...n])
    Merge(A[1...m], A(m+1...n])
```

Lemma

MergeSort *correctly sort the input array.*

Proof by induction on n.

Proving correctness of merge-sort

Lemma

MergeSort *correctly sort the input array.*

```
MergeSort(A[1...n])
    if n ≤ 1 then return
    m ← ⌊n/2⌋
    MergeSort(A[1...m])
    MergeSort(A[m + 1...n])
    Merge(A[1...m], A(m + 1...n])
```

Proof: By induction on n.

Proving correctness of merge-sort

Lemma

MergeSort *correctly sort the input array.*

```
MergeSort(A[1...n])
    if n ≤ 1 then return
    m ← ⌊n/2⌋
    MergeSort(A[1...m])
    MergeSort(A[m + 1...n])
    Merge(A[1...m], A(m + 1...n])
```

Proof: By induction on n.

Base: $n = 1$.

Proving correctness of merge-sort

Lemma

MergeSort *correctly sort the input array.*

```
MergeSort(A[1...n])
    if n ≤ 1 then return
    m ← ⌊n/2⌋
    MergeSort(A[1...m])
    MergeSort(A[m + 1...n])
    Merge(A[1...m], A(m + 1...n])
```

Proof: By induction on n.

Base: $n = 1$.

Inductive hypothesis Lemma correct for all $n \leq k$.

Proving correctness of merge-sort

Lemma

MergeSort *correctly sort the input array.*

```
MergeSort(A[1...n])
    if n ≤ 1 then return
    m ← ⌊n/2⌋
    MergeSort(A[1...m])
    MergeSort(A[m+1...n])
    Merge(A[1...m], A(m+1...n])
```

Proof: By induction on n.

Base: $n = 1$.

Inductive hypothesis Lemma correct for all $n \leq k$.

Inductive step: Need to prove that lemma holds for $n = k + 1 \geq 2$.

Proving correctness of merge-sort

Lemma

MergeSort *correctly sort the input array.*

```
MergeSort(A[1...n])
    if n ≤ 1 then return
    m ← ⌊n/2⌋
    MergeSort(A[1...m])
    MergeSort(A[m + 1...n])
    Merge(A[1...m], A(m + 1...n])
```

Proof: By induction on n.

Base: $n = 1$.

Inductive hypothesis Lemma correct for all $n \leq k$.

Inductive step: Need to prove that lemma holds for $n = k + 1 \geq 2$.

$m = \lfloor n/2 \rfloor < n$: Can use induction on $A[1...m]$.

Proving correctness of merge-sort

Lemma

MergeSort *correctly sort the input array.*

```
MergeSort(A[1...n])
    if n ≤ 1 then return
    m ← ⌊n/2⌋
    MergeSort(A[1...m])
    MergeSort(A[m + 1...n])
    Merge(A[1...m], A(m + 1...n])
```

Proof: By induction on n.

Base: $n = 1$.

Inductive hypothesis Lemma correct for all $n \leq k$.

Inductive step: Need to prove that lemma holds for $n = k + 1 \geq 2$.

$m = \lfloor n/2 \rfloor < n$: Can use induction on $A[1...m]$.

$n - m < n$: Can use induction on $A[m + 1...n]$.

Proving correctness of merge-sort

Lemma

MergeSort *correctly sort the input array.*

```
MergeSort(A[1...n])
    if n ≤ 1 then return
    m ← ⌊n/2⌋
    MergeSort(A[1...m])
    MergeSort(A[m + 1...n])
    Merge(A[1...m], A(m + 1...n])
```

Proof: By induction on n.

Base: $n = 1$.

Inductive hypothesis Lemma correct for all $n \leq k$.

Inductive step: Need to prove that lemma holds for $n = k + 1 \geq 2$.

$m = \lfloor n/2 \rfloor < n$: Can use induction on $A[1...m]$.

$n - m < n$: Can use induction on $A[m + 1...n]$.

$\implies A[1...m], A[m + 1...n]$ are sorted correctly. by induction.

Proving correctness of merge-sort

Lemma

MergeSort *correctly sort the input array.*

```
MergeSort(A[1...n])
    if n ≤ 1 then return
    m ← ⌊n/2⌋
    MergeSort(A[1...m])
    MergeSort(A[m+1...n])
    Merge(A[1...m], A(m+1...n])
```

Proof: By induction on n.

Base: $n = 1$.

Inductive hypothesis Lemma correct for all $n \leq k$.

Inductive step: Need to prove that lemma holds for $n = k + 1 \geq 2$.

$m = \lfloor n/2 \rfloor < n$: Can use induction on $A[1...m]$.

$n - m < n$: Can use induction on $A[m + 1...n]$.

$\implies A[1...m], A[m + 1...n]$ are sorted correctly. by induction.

Since **Merge** is correct $\implies A[1...n]$ is sorted correctly.

THE END

...

(for now)