

Techniques for Algorithm Verification: Array Component Assignment Example

Stefan D. Bruda

CS 310, Winter 2025

- Swapping two variables without using extra space:

```
ASSERT(x==x0 && y==y0)
ASSERT(y == y0 && x - y + y == x0)
x = x - y;
ASSERT(y == y0 && x + y == x0)
ASSERT(x + y - x == y0 && x + y == x0)
y = x + y;
ASSERT(y - x == y0 && y == x0)
x = y - x;
ASSERT(x==y0 && y==x0)
```

- Will the same statement be valid for two indices in an array?

```
ASSERT(A[i]==A0[i] && A[j]==A0[j])
A[i] = A[i] - A[j];
A[j] = A[i] + A[j];
A[i] = A[j] - A[i];
ASSERT(A[i]==A0[j] && A[j]==A0[i])
```

IN-PLACE SWAP, THE ARRAY EDITION



```
ASSERT( (A[i->A[j] - A[i)][i] == AO[j] && (A[i->A[j] - A[i)][j] == AO[i] )  
A[i] = A[j] - A[i];  
ASSERT(A[i]==AO[j] && A[j]==AO[i])
```

IN-PLACE SWAP, THE ARRAY EDITION



```
ASSERT( A[j] - A[i] == AO[j] && ( j == i && A[j] - A[i] == AO[i] ||  
                                     j != i && A[j] == AO[i] )  
ASSERT( (A[i->A[j] - A[i])[i] == AO[j] && (A[i->A[j] - A[i])[j] == AO[i] )  
A[i] = A[j] - A[i];  
ASSERT(A[i]==AO[j] && A[j]==AO[i])
```



IN-PLACE SWAP, THE ARRAY EDITION

```
ASSERT( (A[j->A[i] + A[j]][j] - (A[j->A[i] + A[j]])[i] == AO[j] &&  
        ( j == i && (A[j->A[i] + A[j]][j] - (A[j->A[i] + A[j]])[i] == AO[i] ||  
          j != i && (A[j->A[i] + A[j]][j] == AO[i] )  
A[j] = A[i] + A[j];  
ASSERT( A[j] - A[i] == AO[j] && ( j == i && A[j] - A[i] == AO[i] ||  
                                     j != i && A[j] == AO[i] )  
ASSERT( (A[i->A[j] - A[i)][i] == AO[j] && (A[i->A[j] - A[i)][j] == AO[i] )  
A[i] = A[j] - A[i];  
ASSERT(A[i]==AO[j] && A[j]==AO[i])
```



IN-PLACE SWAP, THE ARRAY EDITION

```
ASSERT( A[i] + A[j] - (A[j->A[i] + A[j])[i] == AO[j] &&
        ( j == i && A[i] + A[j] - A[i] - A[j] == AO[i] ||
          j != i && A[i] + A[j] == AO[i] ) )
ASSERT( (A[j->A[i] + A[j])[j] - (A[j->A[i] + A[j])[i] == AO[j] &&
        ( j == i && (A[j->A[i] + A[j])[j] - (A[j->A[i] + A[j])[i] == AO[i] ||
          j != i && (A[j->A[i] + A[j])[j] == AO[i] ) )
A[j] = A[i] + A[j];
ASSERT( A[j] - A[i] == AO[j] && ( j == i && A[j] - A[i] == AO[i] ||
                                  j != i && A[j] == AO[i] ) )
ASSERT( (A[i->A[j] - A[i])[i] == AO[j] && (A[i->A[j] - A[i])[j] == AO[i] )
A[i] = A[j] - A[i];
ASSERT(A[i]==AO[j] && A[j]==AO[i])
```



IN-PLACE SWAP, THE ARRAY EDITION

```
ASSERT( ( j == i && A[i] + A[j] - A[i] - A[j] == AO[j] ||
        j != i && A[i] + A[j] - A[i] == AO[j] ) &&
        ( j == i && A[i] + A[j] - A[i] - A[j] == AO[i] ||
        j != i && A[i] + A[j] == AO[i] ) )
ASSERT( A[i] + A[j] - (A[j->A[i] + A[j]))[i] == AO[j] &&
        ( j == i && A[i] + A[j] - A[i] - A[j] == AO[i] ||
        j != i && A[i] + A[j] == AO[i] ) )
ASSERT( (A[j->A[i] + A[j]))[j] - (A[j->A[i] + A[j]))[i] == AO[j] &&
        ( j == i && (A[j->A[i] + A[j]))[j] - (A[j->A[i] + A[j]))[i] == AO[i] ||
        j != i && (A[j->A[i] + A[j]))[j] == AO[i] ) )
A[j] = A[i] + A[j];
ASSERT( A[j] - A[i] == AO[j] && ( j == i && A[j] - A[i] == AO[i] ||
        j != i && A[j] == AO[i] ) )
ASSERT( (A[i->A[j] - A[i]))[i] == AO[j] && (A[i->A[j] - A[i]))[j] == AO[i] )
A[i] = A[j] - A[i];
ASSERT(A[i]==AO[j] && A[j]==AO[i])
```

IN-PLACE SWAP, THE ARRAY EDITION

```

ASSERT( ( j == i && O == AO[j] && O == AO[i] ) ||
        ( j != i && A[j] == AO[j] && A[i] + A[j] == AO[i] ) )
ASSERT( ( j == i && O == AO[j] || j != i && A[j] == AO[j] ) &&
        ( j == i && O == AO[i] || j != i && A[i] + A[j] == AO[i] ) )
ASSERT( ( j == i && A[i] + A[j] - A[i] - A[j] == AO[j] ||
        j != i && A[i] + A[j] - A[i] == AO[j] ) &&
        ( j == i && A[i] + A[j] - A[i] - A[j] == AO[i] ||
        j != i && A[i] + A[j] == AO[i] ) )
ASSERT( A[i] + A[j] - (A[j->A[i] + A[j])[i] == AO[j] &&
        ( j == i && A[i] + A[j] - A[i] - A[j] == AO[i] ||
        j != i && A[i] + A[j] == AO[i] ) )
ASSERT( (A[j->A[i] + A[j])[j] - (A[j->A[i] + A[j])[i] == AO[j] &&
        ( j == i && (A[j->A[i] + A[j])[j] - (A[j->A[i] + A[j])[i] == AO[i] ||
        j != i && (A[j->A[i] + A[j])[j] == AO[i] ) )
A[j] = A[i] + A[j];
ASSERT( A[j] - A[i] == AO[j] && ( j == i && A[j] - A[i] == AO[i] ||
        j != i && A[j] == AO[i] ) )
ASSERT( (A[i->A[j] - A[i])[i] == AO[j] && (A[i->A[j] - A[i])[j] == AO[i] )
A[i] = A[j] - A[i];

ASSERT(A[i]==AO[j] && A[j]==AO[i])

```

IN-PLACE SWAP, THE ARRAY EDITION

```

ASSERT( ( j == i && 0 == AO[j] && 0 == AO[i] ) ||
        ( j != i && (A[i->A[i] - A[j]][j] == AO[j] &&
                    (A[i->A[i] - A[j]][i] + (A[i->A[i] - A[j]][j] == AO[i] ) )
A[i] = A[i] - A[j];
ASSERT( ( j == i && 0 == AO[j] && 0 == AO[i] ) ||
        ( j != i && A[j] == AO[j] && A[i] + A[j] == AO[i] ) )
ASSERT( ( j == i && 0 == AO[j] || j != i && A[j] == AO[j] ) &&
        ( j == i && 0 == AO[i] || j != i && A[i] + A[j] == AO[i] ) )
ASSERT( ( j == i && A[i] + A[j] - A[i] - A[j] == AO[j] ||
        j != i && A[i] + A[j] - A[i] == AO[j] ) &&
        ( j == i && A[i] + A[j] - A[i] - A[j] == AO[i] ||
        j != i && A[i] + A[j] == AO[i] ) )
ASSERT( A[i] + A[j] - (A[j->A[i] + A[j])[i] == AO[j] &&
        ( j == i && A[i] + A[j] - A[i] - A[j] == AO[i] ||
        j != i && A[i] + A[j] == AO[i] ) )
ASSERT( (A[j->A[i] + A[j])[j] - (A[j->A[i] + A[j])[i] == AO[j] &&
        ( j == i && (A[j->A[i] + A[j])[j] - (A[j->A[i] + A[j])[i] == AO[i] ||
        j != i && (A[j->A[i] + A[j])[j] == AO[i] )
A[j] = A[i] + A[j];
ASSERT( A[j] - A[i] == AO[j] && ( j == i && A[j] - A[i] == AO[i] ||
        j != i && A[j] == AO[i] ) )
ASSERT( (A[i->A[j] - A[i])[i] == AO[j] && (A[i->A[j] - A[i])[j] == AO[i] )
A[i] = A[j] - A[i];

ASSERT(A[i]==AO[j] && A[j]==AO[i])

```



IN-PLACE SWAP, THE ARRAY EDITION

```

ASSERT( ( j == i && 0 == AO[j] && 0 == AO[i] ) ||
        ( j != i && A[j] == AO[j] && A[i] - A[j] + A[j] == AO[i] ) )
ASSERT( ( j == i && 0 == AO[j] && 0 == AO[i] ) ||
        ( j != i && (A[i->A[i] - A[j])[j] == AO[j] &&
                     (A[i->A[i] - A[j])[i] + (A[i->A[i] - A[j])[j] == AO[i] ) )
A[i] = A[i] - A[j];
ASSERT( ( j == i && 0 == AO[j] && 0 == AO[i] ) ||
        ( j != i && A[j] == AO[j] && A[i] + A[j] == AO[i] ) )
ASSERT( ( j == i && 0 == AO[j] || j != i && A[j] == AO[j] ) &&
        ( j == i && 0 == AO[i] || j != i && A[i] + A[j] == AO[i] ) )
ASSERT( ( j == i && A[i] + A[j] - A[i] - A[j] == AO[j] ||
        j != i && A[i] + A[j] - A[i] == AO[j] ) &&
        ( j == i && A[i] + A[j] - A[i] - A[j] == AO[i] ||
        j != i && A[i] + A[j] == AO[i] ) )
ASSERT( A[i] + A[j] - (A[j->A[i] + A[j])[i] == AO[j] &&
        ( j == i && A[i] + A[j] - A[i] - A[j] == AO[i] ||
        j != i && A[i] + A[j] == AO[i] ) )
ASSERT( (A[j->A[i] + A[j])[j] - (A[j->A[i] + A[j])[i] == AO[j] &&
        ( j == i && (A[j->A[i] + A[j])[j] - (A[j->A[i] + A[j])[i] == AO[i] ||
        j != i && (A[j->A[i] + A[j])[j] == AO[i] )
A[j] = A[i] + A[j];
ASSERT( A[j] - A[i] == AO[j] && ( j == i && A[j] - A[i] == AO[i] ||
        j != i && A[j] == AO[i] ) )
ASSERT( (A[i->A[j] - A[i])[i] == AO[j] && (A[i->A[j] - A[i])[j] == AO[i] )
A[i] = A[j] - A[i];

ASSERT(A[i]==AO[j] && A[j]==AO[i])

```

IN-PLACE SWAP, THE ARRAY EDITION

```

ASSERT( ( j == i && 0 == AO[j] && 0 == AO[i] ) ||
        ( j != i && A[j] == AO[j] && A[i] == AO[i] ) )
ASSERT( ( j == i && 0 == AO[j] && 0 == AO[i] ) ||
        ( j != i && A[j] == AO[j] && A[i] - A[j] + A[j] == AO[i] ) )
ASSERT( ( j == i && 0 == AO[j] && 0 == AO[i] ) ||
        ( j != i && (A[i->A[i] - A[j])[j] == AO[j] &&
                     (A[i->A[i] - A[j])[i] + (A[i->A[i] - A[j])[j] == AO[i] ) )
A[i] = A[i] - A[j];
ASSERT( ( j == i && 0 == AO[j] && 0 == AO[i] ) ||
        ( j != i && A[j] == AO[j] && A[i] + A[j] == AO[i] ) )
ASSERT( ( j == i && 0 == AO[j] || j != i && A[j] == AO[j] ) &&
        ( j == i && 0 == AO[i] || j != i && A[i] + A[j] == AO[i] ) )
ASSERT( ( j == i && A[i] + A[j] - A[i] - A[j] == AO[j] ||
        j != i && A[i] + A[j] - A[i] == AO[j] ) &&
        ( j == i && A[i] + A[j] - A[i] - A[j] == AO[i] ||
        j != i && A[i] + A[j] == AO[i] ) )
ASSERT( A[i] + A[j] - (A[j->A[i] + A[j])[i] == AO[j] &&
        ( j == i && A[i] + A[j] - A[i] - A[j] == AO[i] ||
        j != i && A[i] + A[j] == AO[i] ) )
ASSERT( (A[j->A[i] + A[j])[j] - (A[j->A[i] + A[j])[i] == AO[j] &&
        ( j == i && (A[j->A[i] + A[j])[j] - (A[j->A[i] + A[j])[i] == AO[i] ||
        j != i && (A[j->A[i] + A[j])[j] == AO[i] )
A[j] = A[i] + A[j];
ASSERT( A[j] - A[i] == AO[j] && ( j == i && A[j] - A[i] == AO[i] ||
        j != i && A[j] == AO[i] ) )
ASSERT( (A[i->A[j] - A[i])[i] == AO[j] && (A[i->A[j] - A[i])[j] == AO[i] )
A[i] = A[j] - A[i];

ASSERT(A[i]==AO[j] && A[j]==AO[i])

```



IN-PLACE SWAP, THE ARRAY EDITION

```
ASSERT(A[i]==AO[i] && A[j]==AO[j] && j != i) // necessarily stronger than before
ASSERT( ( j == i && 0 == AO[j] && 0 == AO[i] ) ||
        ( j != i && A[j] == AO[j] && A[i] == AO[i] ) )
ASSERT( ( j == i && 0 == AO[j] && 0 == AO[i] ) ||
        ( j != i && A[j] == AO[j] && A[i] - A[j] + A[j] == AO[i] ) )
ASSERT( ( j == i && 0 == AO[j] && 0 == AO[i] ) ||
        ( j != i && (A[i->A[i] - A[j])[j] == AO[j] &&
                     (A[i->A[i] - A[j])[i] + (A[i->A[i] - A[j])[j] == AO[i] ) )
A[i] = A[i] - A[j];
ASSERT( ( j == i && 0 == AO[j] && 0 == AO[i] ) ||
        ( j != i && A[j] == AO[j] && A[i] + A[j] == AO[i] ) )
ASSERT( ( j == i && 0 == AO[j] || j != i && A[j] == AO[j] ) &&
        ( j == i && 0 == AO[i] || j != i && A[i] + A[j] == AO[i] ) )
ASSERT( ( j == i && A[i] + A[j] - A[i] - A[j] == AO[j] ||
        j != i && A[i] + A[j] - A[i] == AO[j] ) &&
        ( j == i && A[i] + A[j] - A[i] - A[j] == AO[i] ||
        j != i && A[i] + A[j] == AO[i] ) )
ASSERT( A[i] + A[j] - (A[j->A[i] + A[j])[i] == AO[j] &&
        ( j == i && A[i] + A[j] - A[i] - A[j] == AO[i] ||
        j != i && A[i] + A[j] == AO[i] ) )
ASSERT( (A[j->A[i] + A[j])[j] - (A[j->A[i] + A[j])[i] == AO[j] &&
        ( j == i && (A[j->A[i] + A[j])[j] - (A[j->A[i] + A[j])[i] == AO[i] ||
        j != i && (A[j->A[i] + A[j])[j] == AO[i] )
A[j] = A[i] + A[j];
ASSERT( A[j] - A[i] == AO[j] && ( j == i && A[j] - A[i] == AO[i] ||
        j != i && A[j] == AO[i] ) )
ASSERT( (A[i->A[j] - A[i])[i] == AO[j] && (A[i->A[j] - A[i])[j] == AO[i] )
A[i] = A[j] - A[i];
ASSERT(A[i]==AO[j] && A[j]==AO[i])
```



```
ASSERT(A[i]==A0[i] && A[j]==A0[j]    && j != i)
A[i] = A[i] - A[j];
A[j] = A[i] + A[j];
A[i] = A[j] - A[i];
ASSERT(A[i]==A0[j] && A[j]==A0[i])
```

- In place swap of two indices i and j in the same array A will **not** work
 - Carrying on the proof shows that the statement is invalid whenever the two indices to be swapped coincide (that is, $i=j$)
 - Strengthening the pre-condition to forbid this situation will make the statement valid, though whether this is a reasonable pre-condition depends on the code that uses this fragment