Chapter 2

February 10, 2023

1 Chapter 2 Getting Started

1.1 Insertion Sort

[1]:
```
#Figure 2-2
A = [5, 2, 4, 6, 1, 3]

def InsertionSort(A):

    if (n := len(A)) <= 1:
      return

    for i in range(1, n):
            key = A[i]
            #Insert A[i] into the sorted sequence A[1..j-1].
            j = i-1
            while j>=0 and key < A[j]:
                A[j+1] = A[j]
                j -= 1
            A[j+1] = key

InsertionSort(A)
print(A)
```

```
[1, 2, 3, 4, 5, 6]
```

[1]:
```
#Exercise 2.1-1
#From left to right
A = [31, 41, 59, 26, 41, 58]
j=2

for j in range(len(A)):
    key = A[j]
    #Insert A[j] into the sorted sequence A[1..j-1]
    i=j-1
    while i>=0 and (A[i] > key):
        A[i+1] = A[i]
        i=i-1
```

```
        print(A)
    A[i+1] = key

print(A)
```

```
[31, 41, 59, 59, 41, 58]
[31, 41, 41, 59, 41, 58]
[31, 31, 41, 59, 41, 58]
[26, 31, 41, 59, 59, 58]
[26, 31, 41, 41, 59, 59]
[26, 31, 41, 41, 58, 59]
```

[122]:
```
#Exercise 2.1-1 Increasing Insertion sort from right to left

A = [31, 41, 59, 26, 41, 58]

for j in reversed(range(len(A))):
    key = A[j]
    i=j+1
    while i<len(A) and A[i]<key:
        A[i-1] = A[i]
        i=i+1
        print(A)
    A[i-1] = key
print(A)
```

```
[31, 41, 26, 26, 41, 58]
[31, 41, 26, 41, 41, 58]
[31, 41, 26, 41, 58, 58]
[31, 26, 26, 41, 58, 59]
[26, 26, 41, 41, 58, 59]
[26, 31, 41, 41, 58, 59]
```

[123]:
```
#Exercise 2.1-2 Non-Increasing Insertion Sort

A = [5, 2, 4, 6, 1, 3]

j=0

for j in range(len(A)):
    key = A[j]

    i=j-1
    while i>=0 and (A[i] < key):
        A[i+1] = A[i]
        i=i-1
        print(A)
    A[i+1] = key
```

```
print(A)
```

```
[5, 2, 2, 6, 1, 3]
[5, 4, 2, 2, 1, 3]
[5, 4, 4, 2, 1, 3]
[5, 5, 4, 2, 1, 3]
[6, 5, 4, 2, 1, 1]
[6, 5, 4, 2, 2, 1]
[6, 5, 4, 3, 2, 1]
```

[84]:
```
#Exercise 2.1-3

A = [1,2,'v',4,5,6,7]

for j in range(len(A)):
    if A[j] == 'v':
        i=j
        print(i)
    else:
        i='NIL'
        print(i)
```

```
NIL
NIL
2
NIL
NIL
NIL
NIL
```

[221]:
```
#Exercise 2.1-4

def AddBinary(A, B):

    carry = 0
    n = max(len(A), len(B))
    C = [0 for i in range(n+1)]

    for i in reversed(range(n)):
        a = A[i] if i < len(A) else 0
        b = B[i] if i < len(B) else 0
        C[i+1] = (a + b + carry) % 2
        carry = (a + b + carry) // 2
    C[0] = carry

    return C
```

```
#Test 0-10

print(AddBinary([0], [0]))
print(AddBinary([0], [1]))
print(AddBinary([1], [1]))
print(AddBinary([1,0], [0,1]))
print(AddBinary([1,0], [1,0]))
print(AddBinary([1,1], [1,0]))
print(AddBinary([1,1], [1,1]))
print(AddBinary([0,1,0], [1,0,1]))
print(AddBinary([0,0,1], [1,1,1]))
print(AddBinary([1,0,0,0], [0,0,0,1]))
print(AddBinary([1,0,0,0], [0,0,1,0]))
```

```
[0, 0]
[0, 1]
[1, 0]
[0, 1, 1]
[1, 0, 0]
[1, 0, 1]
[1, 1, 0]
[0, 1, 1, 1]
[1, 0, 0, 0]
[0, 1, 0, 0, 1]
[0, 1, 0, 1, 0]
```