http://www.cs.cornell.edu/courses/cs1110/2018sp

Lecture 26: Sorting

CS 1110

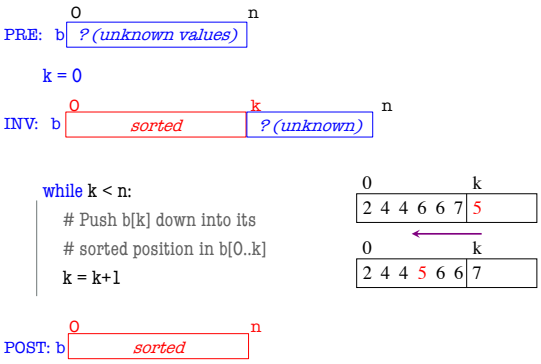
Introduction to Computing Using Python

[E. Andersen, A. Bracy, D. Gries, L. Lee, S. Marschner, C. Van Loan, W. White]

Plan of Attack

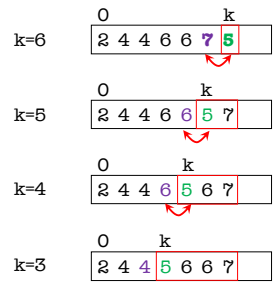
- **Insertion Sort**
- Partition
- Quick Sort

Insertion Sort



Insertion Sort: Moving into Position

```
def push_down(b, k):
    while k > 0:
        if b[k-1] > b[k]:
            swap(b,k-1,k)
        k = k-1

k = 0
while k < n:
    push_down(b,k)
    k = k+1
```



Is this how you want to sort 500 exams?

Algorithm Complexity

```
def push_down(b, k):
    while k > 0:
        if b[k-1] > b[k]:
            swap(b,k-1,k)
        k = k-1

k = 0
while k < n:
    push_down(b,k)
    k = k+1
```

Nested loops multiply the number of operations required. We need to compare b[k] to all n elements. *n* operations

Iterating through a sequence of length *n* requires *n* operations: push_down called *n* times

Q: Algorithm Complexity

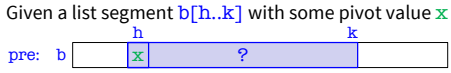
```
def push_down(b, k):
    while k > 0:
        if b[k-1] > b[k]:
            swap(b,k-1,k)
        k = k-1

k = 0
while k < n:
    push_down(b,k)
    k = k+1
```

Approximately how many operations does this take?

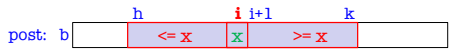
A: ~ 1 operation
B: ~ n operations
C: ~ n^2 operations
D: ~ n^3 operations
E: I don't know

Partition Algorithm

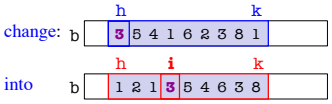
- Given a list segment b[h..k] with some pivot value x in b[h]:



- Swap elements of b[h..k] and store in i to satisfy postcondition:



Example:

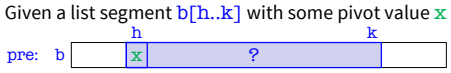


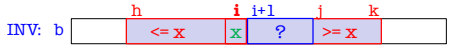
x
- Called the pivot value
- not a variable

= whatever value is in b[h]

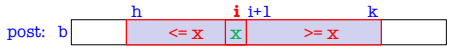
Partition: What's the Invariant?

- Given a list segment b[h..k] with some pivot value x in b[h]:





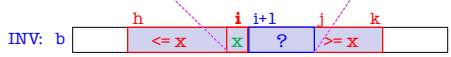
- Swap elements of b[h..k] and store in i to satisfy postcondition:



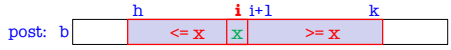
Partition: What's the Invariant?

- Given a list segment b[h..k] with some pivot value x in b[h]:

Initially i = h, j = k+1

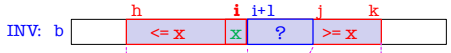


- Swap elements of b[h..k] and store in i to satisfy postcondition:



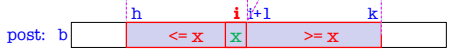
Partition: What's the Invariant?

- Given a list segment b[h..k] with some pivot value x in b[h]:

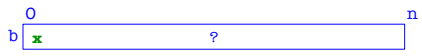


Eventually j = i+1

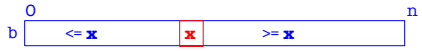
- Swap elements of b[h..k] and store in i to satisfy postcondition:



Sorting with Partitions



- **Idea:** Pick a *pivot* element **x**
- Partition sequence into <= x and >= x



Now Partition this and this, too

Keep recursing…

QuickSort

```
def quick_sort(b, h, k):
    """Sort the array fragment b[h..k]"""
    if k<=h:
        return
    i = partition(b, h, k)
    # INV: b[h..i-1] <= b[i] <= b[i+1..k]
    # Sort b[h..i-1] and b[i+1..k]
    quick_sort (b, h, i-1)
    quick_sort (b, i+1, k)
```