

**Data Structures with Python
(20CS41P)
Important Questions for Practical Exams**

(1) Basic Data Structure

Ans:-

Algorithms:-

Step 1:[Input Operation & Import time]

```
import time
```

```
start = time.time()
```

```
List=[1,2,"ABC", 3, "xyz", 2.3]
```

```
Dict={"a":1,"b":2,"c":3}
```

```
Tup=(1,2,3,4,5)
```

```
S={1,1,2,2,3,3,4,4,5,5}
```

Step 2:[Output operation]

```
Print list("List")
```

```
Print ("\nDictionary")
```

```
Print ("\n Tuples")
```

```
Print ("\n Sets")
```

```
Print(list)
```

```
Print (Dict)
```

```
Print (Tup)
```

```
Print (s)
```

```
end = time.time()
```

```
print(f"Runtime of the program is {end - start}")
```

Python Code:-

```
import time
```

```
start = time.time()print("List")
```

```
List = [1, 2,"ABC", 3, "xyz", 2.3]
```

```
print(List)
```

```
print("\nDictionary")
```

```
Dict={"a":1,"b":2,"c":3}
```

```

print(Dict)
print("\n Tuples")
Tup=(1,2,3,4,5)
print (Tup)
print("\n Sets")
s={1,1,2,2,3,3,4,4,5,5}
print(s)
end = time.time()
print(f"Runtime of the program is {end – start}")

```

(2) Linear Search

Ans:-

Linear search algorithm:-

Step 1: [Import time]

Import time

start=time.time()

Step 2: [Define a function for linear search]

```
def linearsearch(a, key):
```

```
n = len(a)
```

```
for i in range(n):
```

```
if a[i] == key:
```

```
return i;
```

```
return -1
```

Step 3: [Define an array]

```
a = [13,24,35,46,57,68,79]
```

```
print("the array elements are:",a)
```

Step 4: [Enter the element to be searched]

```
k = int(input("enter the key element to search:"))
```

Step 5: [Output operation]

```
i = linearsearch(a,k)if i == -1:
```

```
print("Search UnSuccessful")
```

```
else:
```

```
print("Search Successful key found at location:",i+1)
```

```
end=time.time()
```

```
print(f"Runtime of the program is {end – start}")
```

Python Code

```
import time
start=time.time()
def linearsearch(a, key):
    n = len(a)
    for i in range(n):
        if a[i] == key:
            return i;
    return -1
a = [13,24,35,46,57,68,79]
print("the array elements are:",a)
k = int(input("enter the key element to search => "))
i = linearsearch(a,k)
if i == -1:
    print("Search UnSuccessful")
else:
    print("Search Successful key found at location:",i+1)
end=time.time()
```

```
print(f"Runtime of the program is {end - start}")
```

(3) Bubble Sort

Ans:-

Bubble sort Algorithm:-

Step 1: [Import time and define a function for Bubble sort]

```
import time
start=time.time()
def bubblesort(a):
    n = len(a)
    for i in range(n-2):
        for j in range(n-2-i):
            if a[j]>a[j+1]:
```

```
temp = a[j]
a[j] = a[j+1]
a[j+1] = temp
Step 2: [Create array and do the operation]
alist = [34,46,43,27,57,41,45,21,70]
Step 3: [Output operation ]
print("Before sorting:",alist)
bubblesort(alist)
end=time.time()
print(f"Runtime of the program is {end – start}")
```

Python Code:-

```
import time
start=time.time()
def bubblesort(a):
    n = len(a)
    for i in range(n-2):
        for j in range(n-2-i):
            if a[j]>a[j+1]:
                temp = a[j]
                a[j] = a[j+1]
                a[j+1] = temp
alist = [34,46,43,27,57,41,45,21,70]
print("Before sorting:",alist)
bubblesort(alist)
print("After sorting:",alist)
end=time.time()
print(f"Runtime of the program is {end - start}")
```

Output:

```
Before sorting: [34, 46, 43, 27, 57, 41, 45, 21, 70]
After sorting: [21, 27, 34, 41, 43, 45, 46, 57, 70]
Runtime of the program is 0.06250619888305664
```

(4) Selection Sort

Ans:-

Selection Sort Algorithm:-

Step 1: [Import time & create function of selection sort]

```
import time
start=time.time()
def selectionsort(a):
    n = len(a)
    for i in range(n-2):
        min = i
        for j in range(i+1,n-1):
            if a[j]<a[min]:
                temp = a[j]
                a[j] = a[min]
                a[min] = temp
```

Step 2: [Define array & execute the operation]

```
alist = [34,46,43,27,57,41,45,21,70]
print("Before sorting:",alist)
selectionsort(alist)
print("After sorting:",alist)
end=time.time()
print(f"Runtime of the program is {end - start}")
```

Python Code:-

Python Code

```
import time
start=time.time()
def selectionsort(a):
    n = len(a)
    for i in range(n-2):
        min = i
        for j in range(i+1,n-1):
            if a[j]<a[min]:
                temp = a[j]
                a[j] = a[min]
                a[ min] = temp
alist = [34,46,43,27,57,41,45,21,70]
print("Before sorting:",alist)
selectionsort(alist)
print("After sorting:",alist)
```

```
end=time.time()
print(f"Runtime of the program is { end - start }")
```

(5) Insertion Sort

Ans:-

Insertion sort Algorithm:-

Step 1: [Define insertion sort long with importing time]

```
import time
start=time.time()
def insertionsort(a):
    n = len(a)
```

```

for i in range(1,n-1):
    k = a[i]
    j = i-1
    while j>=0 and a[j]>k:
        a[j+1] = a[j]
        j=j-1
    a[j+1] = k
Step 2: [Create an array]
alist = [34,46,43,27,57,41,45,21,70]
print("Before sorting:",alist)
Step 3: [Output operation]
insertionsort(alist)
print("After sorting:",alist)end=time.time()
print(f"Runtime of the program is {end – start}")

```

Python code

```

import time
start=time.time()
def insertionsort(a):
    n = len(a)
    for i in range(1,n-1):
        k = a[i]
        j = i-1
        while j>=0 and a[j]>k:
            a[j+1] = a[j]
            j=j-1
        a[j+1] = k
alist = [34,46,43,27,57,41,45,21,70]
print("Before sorting:",alist)
insertionsort(alist)
print("After sorting:",alist)
end=time.time()
print(f"Runtime of the program is {end - start}")

```

(6) Fibonacci

Ans:-

Algorithm:-

Step 1: [Defining Fibonacci function & import time]

```
import time
```

```
start = time.time()
```

```
def fibonacci(n):
```

```
# Taking 1st two fibonacci numbers as 0 and 1
```

```
f = [0, 1]
```

```
for i in range(2, n+1):
```

```
f.append(f[i-1] + f[i-2])
```

```
return f[n]
```

Step 2: [Output Operation]print(fibonacci(9))

```
end = time.time()
```

```
print(f"Runtime of the program is {end - start}")
```

Python Code

```
import time
```

```
start = time.time()
```

```
def fibonacci(n):
```

```
    f = [0, 1]
```

```
    for i in range(2, n+1):
```

```
        f.append(f[i-1] + f[i-2])
```

```
    return f[n]
```

```
print(fibonacci(9))
```

```
end = time.time()
```

```
print(f"Runtime of the program is {end - start}")
```


(7) Implement Stack Data

Ans:-

Algorithm:-

1. Implement Stack Data

Step 1: [Create stack of list & import time]

```
import time
```

```
start = time.time()
```

```
stack = []
```

Step 2: [Function to push element in the stack]

```
stack.append('a')
```

```
stack.append('b')
```

```
stack.append('c')
```

```
print('Initial stack')
```

```
print(stack)
```

Step 3: [Function to pop element from stack]

```
print(stack.pop())
```

```
print(stack.pop())
```

```
print(stack.pop())
```

Step 4: [Output operation]

```
print('\nElements popped from stack:')
```

```
print('\nStack after elements are popped:')
```

```
print(stack)
```

```
end = time.time()
```

```
print(f"Runtime of the program is {end - start}")
```

Python Code

```
import time
start = time.time()
stack = []
stack.append('a')
stack.append('b')
stack.append('c')
print('Initial stack')
print(stack)
print("\nElements popped from stack:")
print(stack.pop())
print(stack.pop())
print(stack.pop())
print("\nStack after elements are popped:")
print(stack)
end = time.time()
print(f"Runtime of the program is {end - start}")
```

(8) Factorial (Recursive)

Ans:-

b) Fibonacci

```
def fib(n):
    if n<=1:
        return n
    return fib(n-1) + fib(n-2)
n=int(input("Enter the range:"))
print("The fibonacci value is:",fib(n))
```

(9) Implement Hash Functions

Ans:-

Algorithm:-

Step 1: [Define variables of different datatypes & import time]

```
import time
```

```
start = time.time()
```

```
int_val = 4
```

```
str_val = 'GeeksforGeeks'
```

```
flt_val = 24.56
```

Step 2: [Print the variables]

```
print("The integer hash value is : " + str(hash(int_val)))
```

```
print("The string hash value is : " + str(hash(str_val)))
```

```
print("The float hash value is : " + str(hash(flt_val)))
```

```
end = time.time()
```

```
print(f"Runtime of the program is {end - start}")
```

Python Code:-

```
import time
```

```
start = time.time()
```

```
int_val = 4
```

```
str_val = 'GeeksforGeeks'
```

```
flt_val = 24.56
```

```
print("The integer hash value is : " + str(hash(int_val)))
```

```
print("The string hash value is : " + str(hash(str_val)))
```

```
print("The float hash value is : " + str(hash(flt_val)))
```

```
end = time.time()
```

```
print(f"Runtime of the program is {end - start}")
```

(10) Array Implementation

Ans:-

```
import array as arr
a = arr.array('i', [2, 4, 5, 6])
print("First element is:", a[0])
print("Second element is:", a[1])
print("Third element is:", a[2])
print("Forth element is:", a[3])
print("last element is:", a[-1])
print("Second last element is:", a[-2])
print("Third last element is:", a[-3])
print("Forth last element is:", a[-4])
print(a[0], a[1], a[2], a[3], a[-1], a[-2], a[-3], a[-4])
```

Output:

Now we compile the above code in python, and after

```
First element is: 2
Second element is: 4
Third element is: 5
Forth element is: 6
last element is: 6
Second last element is: 5
Third last element is: 4
Forth last element is: 2
2 4 5 6 6 5 4 2
```

Remaining 2 programs name:-

(11) Singly Linked List

(12) Breadth First Search

It's there in Group

Read It