First homework of algorithm design

Related master : Dr . Marzieh Rahimi

Teaching Assistants :

1 . Hamed Yahyazadeh

2. Sahar Harati

Deadline for submission :

7’th April 2021(1400/1/18)

1 . for each function f(n) and time t in the following table, determine the largest size n of a problem that can be solved in time t, assuming that the algorithm to solve the problem takes f(n) microseconds.

	1 second	1 minute	1 hour	1 day	1 month	1 year	1 century
$\lg n$							
$\sqrt{n}$							
n							
$n \lg n$							
n^2							
n^3							
2^n							
$n!$							

(*guidance: To find the largest size of problem that can be solved in time t, we need to solve the following equation for n :

$f(n) = t$ **in microseconds**

2 . find the following codes' time complexity.

*** just write the time complexity like this : o(n^2)**

A)
```
for (i = o; i <= n - 1; i++)
for (j = 1; j <= n - i; j++)
a[j][i + j] = i;
```

B)
```
for (i = 1; i <= n; i++)
for (j = 1; j <= i; j++)
for (k = 1; k <= n; k++)
x = x + 1;
```

C)
```
for (i = 1; i <= n; i++){
for (i = 1; i <= n; i++)
k = k + 1;
j = 1;
while (j < n){
k = k + 1;
j = 2 * j;
    }
}
```

D)
```
i = n;
while( i >= 1){
for (j = 1; j <= n; j++)
x = x + 1;
i = i / 3;
}
```

E)
```
for (int i = n; i > 0; i = sqrt(i)) {
// some O(1) expressions
}
```

```
F) for (int i=1; i<=n; i++)
{
int p = pow(i, k);
for (int j=1; j<=p; j++)
{
// Some O(1) work
}
}
```

3 . You know that in the worst case, Merge Sort works better than insertion sort. But small problems of insertion sort can be faster in practice than the Merge Sort, so in practice It can be possible to improve the time complexity (running time) by solving sub-problems in the Merge Sort by Insertion Sort suppose that we want to organize $\frac{n}{k}$ Small sub-problems with length k in Merge Sort with Insertion Sort, explain the related algorithm and calculate It's time complexity. (OK?)

توضیح : ترکیب مرتب سازی درجی و ادغام برای بهبود زمان اجرا.

4 . Reverse : the array of A[1…n] is an array of *n* different element. We call (i , j) a reverse if :

for any (i , j) , | ***i < j and a[i] > a[j]***

Find 5 Reverses of the following array :

[2,3,8,6,1]

5 .

A. Explain all four steps of Heap Sort Algorithm coming below . (at most on one page)

1. **Build-MaxHeap(A)**
2. **for ⟵ length (A) down to 2**
3. **do exchange A[1] ⟷ A[i]**
4. **Max-heapify (A , 1 , i – 1)**

B. Please illustrate the operation of Heap Sort on the following Array :

4	**28**	**11**	**2**	**35**	**12**	**9**	**35**	**16**

C. What is the running time of Heap Sort?

D. What is the worst case running time of Heap Sort ? (Why?)

E. Implement Heap Sort using java programming language and send It to quera. (send your file please)

F. What are the advantages and disadvantages of Heap Sort?

G. Explain the memory usage of Heap Sort. (at most on 5 lines)

Scoring questions

1 . **Implement the Merge Sort and Insertion Sort using C# programming language and send It to me on telegram PV (by your name and ID).**

2 .

a) **search about divide and conquer algorithm.**
b) **explain about binary search with divide and conquer algorithm.**

*** For a and b explain them at most on one page.**

c) **Implement binary search with divide and conquer algorithm using C# programming language and send It to me on telegram PV(by your name and ID)**

Telegram PV : @elebargh

Security warning : Do not copy your Homework.

Best wishes