CSE 431/531 Homework 1 Spring 2022

Feng Wei 50428109

Problem 1.

$f(n)$	$g(n)$	O	Ω	Θ
$\log_2 n$	$5\log_2(n^3)+3$	Yes	Yes	Yes
$10n^2-n$	$n^2\log n$	Yes	no	no
n^3-4n^2+10	n^2	no	Yes	no

prove $\lceil 10n\sqrt{n}\rceil + \lceil n\log n\rceil =$ [redacted] $O(n\sqrt{n})$

Solution: $\lceil 10n\sqrt{n}\rceil + \lceil n\log n\rceil$

$\leq (10n\sqrt{n}+1) + (n\log n+1)$

$\leq 10n\sqrt{n} + n\log n + 2$

Since $\log n = 2\log\sqrt{n} \leq 2\sqrt{n}$ for every $n>0$

$\leq 10n\sqrt{n} + 2n\log\sqrt{n} + 2$

$\leq 10n\sqrt{n} + 2n\sqrt{n} + 2$

$\leq 12n\sqrt{n} + 2$

$= O(n\sqrt{n})$

So: $\lceil 10n\sqrt{n}\rceil + \lceil n\log n\rceil = O(n\sqrt{n})$

· problem 2.

(2a) the pseudo code exchanges the values of A[i] and A[j].

(2b) The running time for the iteration i of the outer loop is:

$$n - i$$

$$\text{Total running time} = \sum_{i=1}^{n-1} (n-i) = (n-1) + (n-2) + \cdots (1)$$

$$= \frac{n(n-1)}{2}$$

Since $\frac{n(n-1)}{2} \leq \frac{1}{2}n^2 = O(n^2)$

$\frac{n(n-1)}{2} = \frac{1}{4}n^2 + \frac{n}{2}(\frac{n}{2} - 1)$ for $n>2$

$\frac{n(n-1)}{2} \geq \frac{1}{4}n^2 = \Omega(n^2)$ $\boxed{\text{Hence the tight bound is } \Theta(n^2)}$

(2C) Correctness: the outer iteration i will select the ith smallest element and put it in the ith place. While the elements before i are already sorted. So, after all the iterations the Array will be sorted.

For example: if $i=1$, then the inner loop will find the smallest element and put it in the ith (1st) place.

if $i = k$ ($<n$), the the elements before k place have been sorted. The inner loop will find the kth smallest element and put it in the kth place.

problem 3.

```
let d_v ← 0 for every v ∈ V
for every v ∈ V do
    for every u such that (v, u) ∈ E do
        d_u ← d_u + 1
S ← {v : d_v = 0}, i ← 0, flg ← Unique
while S ≠ ∅ do
    v ← arbitrary vertex in S, S ← S \ {v}
    i ← i + 1, π(v) ← i, choice = 0
    for every u such that (v, u) ∈ E do
        d_u ← d_u - 1
        if d_u = 0 then add u to S, choice ← choice + 1
    if choice > 1 then flg ← multiple

if i < n then output "none"
else if flg = Unique then output "Unique"
else output "multiple"
```

Running time: For Algorithm line 1-6 every vertex v is handled once. When handle v, the running time is the outgoing edges of v. So the total time of line 1-6 is $\sum_{v \in V} O(1+d^{out}(v))$

$$= O(n+m)$$

For Algorithm line 6 - 12, in the while loop every vertex is handled at most once. The running time for line 9 is the outgoing edges of v.

So the total running time is $\sum_{v \in V} O(1+d^{out}(v))$

$$= O(n+m)$$

Correctness: For most of the Algorithm, I followed the instruction from the slides. So the correctness is obvious.

I added a flag at line 5, which indicates there is unique or multiple topological ordering. At line 11, if there are more than one u for a vertex v that $d_u = 0$, the I set the choice ← choice + 1, That is we have multiple choices. So I set the flag to multiple.

At line 13, if $i < n$, there means we can not put all the vertex in a topological ordering.

At line 14, we can put all the vertex in a topological ordering at the same time flg = unique, then the topological ordering is unique.

At line 15, we can put all the vertex in a topological ordering while flg = multiple. then the output is "multiple"