

## HW2 Yuso Xing

1. a. runtime complexity is

for  $f()$ : there is a loop to run from 0 to  $n-1$ .

for  $g()$ : there is a loop too to run from 0 to  $n$ .

12. int  $h(\text{int } n)$

```
{ return  $n * (n-1)/2$  ;  
}
```

2. for the  $g(n)$ 's runtime complexity.

there is a for loop, from 1 to  $n-1$ .

The time for log. is  $\log(n)$

so the time complexity is  $\log(n) \cdot \log(n) = (\log(n))^2$



3. ① Nonnegative integer as input

②  $x^n$  as output

Procedure;

if  $n=0$  then  $y \leftarrow 1$

else  $y \leftarrow \text{power}(x, (n/2))$

$y \leftarrow y^2$

if  $n$  is odd then  $y \leftarrow xy$

end if

return  $y$

Yes.



4. a. add even: odd: if  $n \% 2 == 0$  => "even"  
else => "odd"  
complexity  $O(1)$

b. exist or not exist: if  $n == m$  => "exist"  
else => "not exist"  
complexity  $O(1)$  (need to go over)

c. smallest number: if  $n < \text{min}$  (min is smallest number)  
min = n;  
complexity  $O(n)$

d. for  $i$  in  $(0, n)$   
for  $j$  in  $(0, m)$   
if  $A[i] != B[j]$ ;  
=> "not same"  
else  
=> "same"  
complexity  $O(n^2)$

e. shorter d:  
for  $i$  in  $(0, n)$   
if  $A[i] != B[i]$   
=> "not same"  
else => "same" (complexity  $O(n)$ )

f. if  $\text{BSI}(n) == 1$  (BSI is the binary search tree function)  
"in"  
else "not in"



5. 111.

include <iostream>

using namespace std;

bool anagram ( string s1, string s2 )

{ if ( s1.length() != s2.length() )

return false;

no ->

int n[256] = {0};

int m[256] = {0};

int i

loop ->

for ( i=0; i < s1.length(); i++ )

{ n[ s1[i] ] ++;  
m[ s2[i] ] ++; }

for ( i=0; i < 256; i++ )

{ if ( n[i] != m[i] )

return false;

else  
return true; }

}

Check

main()

{ cout << anagram  
"yes", "yes"?  
endl; }

Time complexity  $O(n)$