ALG2 Assignment A

Answer the questions in the document!

1. The following bidirectional graph G = (V,E) with V = {1,2,3,4,5,6,7,8,9} is given with an adjacency matrix:

E =

0	1	1	0	0	0	0	0	0
1	0	1	0	0	1	0	0	0
1	1	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	1	1	0	0	1	0	0
0	1	0	0	0	0	1	0	0
0	0	0	0	1	1	0	1	1
0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0

You want to colour the vertices of G with either red or green such that all edges have at least one red vertex. Of course one simple solution is to colour all vertices red. But we assume that colouring vertices red is much more expensive then colouring them green

   a. What is the minimum number of vertices that needs to be coloured red such that the above given solution is satisfied? 4
   b. Which vertices should be coloured red? 1, 2, 3, 7

2. Does the speed of the computer can influence the complexity of an algorithm?
   a. Yes
   b. No
   c. It depend on the algorithm
   d. It depends on the implementation of the algorithm

3. Given the following code

```
s=5;
while s > 0:
    s = s-1
```

How many “simple” operations (timesteps) are executed when running this code? 10

4. We consider the following code that runs for a given input:

INPUT: string s[0]….s[n-1] # a string with length n

```
count = 0;
for character in s:
    if character == ‘a’ :
        count = count +1
```

a. How many “simple” operation are executed by this code on a string with length n and with a number of occurrences of ‘a’ on the input string s? n + amount of a
b. What is the best possible non-empty input? a single letter that is not A
c. What is the worst possible input? And how many “simple” operations are executed in this case? only a's, we have 2n operations

5. You are making an application that makes use of an algorithm A. When you analyse the time performances of your application which analysis of the algorithm A you need to take into account:
   a. The best case running time
   b. The worst case running time
   c. The average running time

6. We consider the following code that runs for a given input:

INPUT: string s[0]….s[n-1] **# a string with length n**

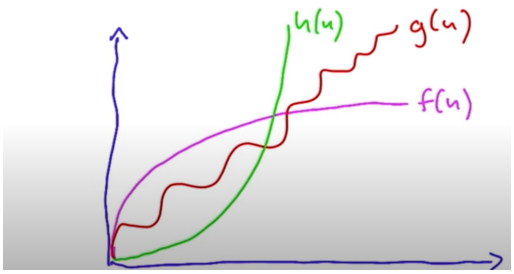
```
count = 0;
for i: 0<= i <= n-1:
    if s[i] == ‘a’:
      if s[i+1] == ‘b’:
         count = count +1
```

We consider the following input sequences as given in the table below (the length of the input sequences is irrelevant) and how they influence the running time of the code. Fill in the table.

Input string	Best case	Worst case	In between
abababab….			
aaaaaaaaa….			
acacacacac….			
bbbbbbbbb….			

?

7. We consider the three functions f(n), g(n) and h(n) as shown in the image below, each expressing a complexity of an algorithm.



Which of the following statements are true:
a. f(n) $\in O(g(n))$

b. $f(n) \in O(h(n))$
c. $g(n) \in O(f(n))$
d. $g(n) \in O(h(n))$
e. $h(n) \in O(f(n))$
f. $h(n) \in O(g(n))$

8. Fill in the table

statement	is it correct?	is it the best possible bound?
$4n^2 - 300n + 12 \in O(n^2)$	yes	yes
$4n^2 - 300n + 12 \in O(n^3)$	yes	no
$3^n + 5n^2 + 3n \in O(n^2)$	yes	yes
$3^n + 5n^2 + 3n \in O(3^n)$	no	
$3^n + 5n^2 + 3n \in O(4^n)$	no	
$50 \cdot 2^n \cdot n^2 + 5n - \log(n) \in O(2^n)$	no	
$50 \cdot 2^n \cdot n^2 + 5n - \log(n) \in O(2.1^n)$	no	
$50 \cdot 2^n \cdot n^2 + 5n - \log(n) \in O(2^n \cdot n^3)$	yes	yes

9. What is the running time of

INPUT: string s[0]….s[n-1] **# a string with length n**

```
count = 0;
for i: 0<= i <= n-1:
    if s[i] == 'a':
      if s[i+1] == 'b':
         count = count +1
```

O(n)

expressed in the Big O notation?

10. What is the running time of the following code expressed in the Big O notation?

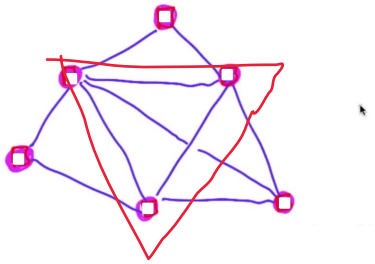
```
result = 0
for i in range(0,n):
   for j in range(0, n):
       result = result +j
```

a. O(n)
b. $O(n^{1.5})$
c. $O(n^2)$
d. $O(n^3)$

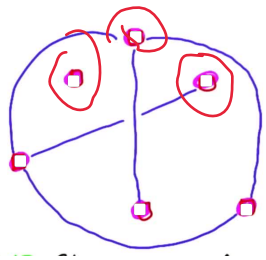
11. Is it a polynomial or exponential running time?

running time	polynomial	exponential
$O(2^n \cdot \log(n))$		X
$O(2^{\log(n)})$	X	
$O(1.00001^n)$		X
$O(n^{1000})$	X	
$O(2^n \cdot n^2)$		X

12. What is the largest clique of the graph below?



13. What is the largest Independent set of the graph below?



14. What do Clique, Vertex cover and Independent Set have in common:
    a. So far, no polynomial algorithm
    b. No practical relevance
    c. "Simple" algorithms go through exponential number of solutions
    d. "Easy" to figure out if a given 0,1 mapping is valid

15. Describe the following terms
    a. decision problem
    b. optimization problem
    c. undecidability problem

a. Decision Problem
A decision problem asks for a yes or no answer based on given inputs, like determining if a number is prime.

b. Optimization Problem
An optimization problem seeks the best solution, such as minimizing costs or maximizing efficiency, within given constraints.

c. Undecidability Problem
An undecidability problem is a problem for which no algorithm can provide a solution for all possible inputs, like the Halting Problem, which asks whether a program will ever stop running.