Laboratory practice No. 5: Graphs

Simón Álvarez Ospina
Universidad EAFIT
Medellín, Colombia
salvarezo1@eafit.edu.co

David Madrid Restrepo
Universidad EAFIT
Medellín, Colombia
dmadridr@eafit.edu.co

November 9, 2020

1 Report

i. For the code in 1.1 see [1]. [2]

ii. For the code in 2.1 see [3]. [4]

iii. *.

i. *.

ii. To calculate the complexity in memory of the code 1.1, we first define n as the number of vertices. Then we say that:

M(n) = $c_1 + c_2 \cdot n + n^2$, where c_1 and c_2 are constants.

Solving the equation, we get the following:

M(n) = n^2

With this last equation, we can define the complexity as: $O(n^2 + c_1 + c_2 \cdot n)$. Using the addition rule, we'll get $O(n^2 + c_2 \cdot n)$, and by the product rule, the complexity is $O(n^2 + n)$. Because $O(n^2) > O(n)$ we say that the final complexity for this algorithm is $O(n^2)$.

iii. *.

iv. *.

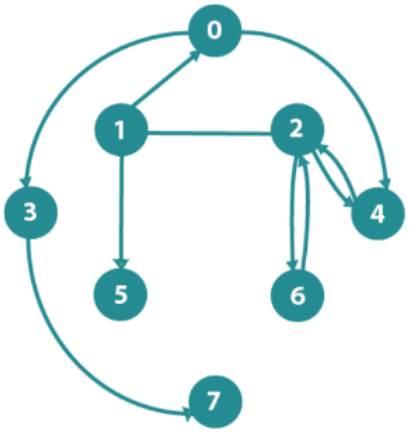
v. To calculate the complexity of this algorithm, first we will say that n and m are the number of nodes and the connections that each node has respectively, thus, we can define the time as follows:

$T(n, m) = c_1 + c_2 \cdot n + c_3 \cdot n \cdot m + c_4 \cdot n^2 \cdot m$, where c_1, c_2, c_3 and c_4 are constants.

We can define T(n,m) as O(n,m), therefore, $O(n, m) = O(c_1 + c_2 \cdot n + c_3 \cdot n \cdot m + c_4 \cdot n^2 \cdot m)$. By the addition rule, factorizing and the product rule, we'll obtain that the complexity is $O(n^2 \cdot m)$.

Professor Mauricio Toro Bermudez
Phone: (+57)(4)2619500 Ext. 9473. Office: 19 – 627
E-mail: mtorobe@eafit.edu.co

2 Midterm Exam [5]



i.

	0	1	2	3	4	5	6	7
0				1	1			
1	1		1			1		
2		1			1		1	
3								1
4			1					
5								
6			1					
7								

ii. **i.** $0 \Rightarrow [3, 4]$

ii. $1 \Rightarrow [0, 2, 5]$

iii. $2 \Rightarrow [1, 4, 6]$

iv. $3 \Rightarrow [7]$

v. $4 \Rightarrow [2]$

vi. $6 \Rightarrow [2]$

iii. b) $O(n^2)$.

iv. **i.** ii. $1, 4, 5, 0, 2, 3.$

ii. i. $1, 4, 5, 0, 2, 3.$

Professor Mauricio Toro Bermudez
Phone: (+57)(4)2619500 Ext. 9473. Office: 19 – 627
E-mail: mtorobe@eafit.edu.co

References

[1] S. Álvarez and D. Madrid. (). "For code 1.1," [Online]. Available: https://github.com/dmadridr/ST0245-002/tree/master/laboratorios/lab05/codigo.

[2] S. Vaid. (). "For the dijkstra algorithm used in 1.1 see," [Online]. Available: https://www.geeksforgeeks.org/shortest-path-unweighted-graph/?ref=lbp.

[3] S. Álvarez and D. Madrid. (). "For code 2.1," [Online]. Available: https://github.com/dmadridr/ST0245-002/blob/master/laboratorios/lab05/codigo/TwoColors.java.

[4] M. Toro. (). "For the digraph code used in 1.1 and 2.1 see," [Online]. Available: https://github.com/mauriciotoro/ST0245-Eafit/tree/master/talleres/taller11/codigo_estudiante/java.

[5] ——, "Laboratorio Nro. 5 Grafos.," version 19., pp. 12–22, 2019. [Online]. Available: https://github.com/mauriciotoro/ST0245-Eafit/blob/master/laboratorios/lab05/ED1-Laboratorio5%20Vr%208.0.pdf.

Professor Mauricio Toro Bermudez
Phone: (+57)(4)2619500 Ext. 9473. Office: 19 – 627
E-mail: mtorobe@eafit.edu.co