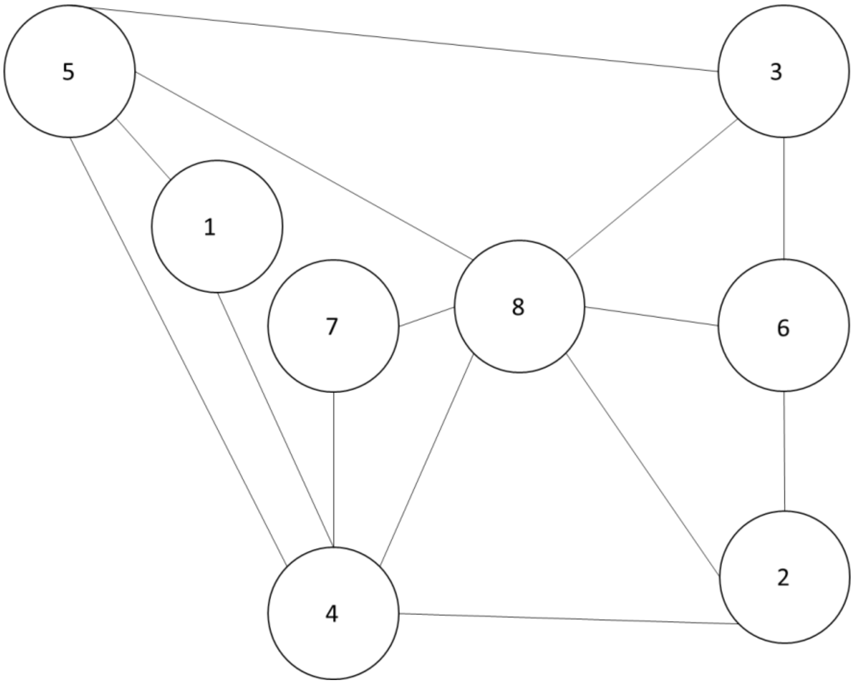
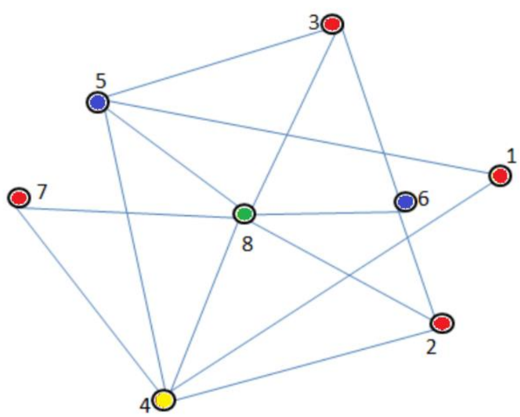
Connor Stewart
ID: 101041125
COMP 1805

Assignment 4

1. Yes, this graph can be represented as a planar graph (no overlapping edges):



2. The graph in question one with the minimum number of colours is as follows:



There are four colours:

Red

Blue

Green

Yellow

Proof of minimum colours by contradiction:

If 1 is changed to blue or yellow it will contradict with 5 (blue) and 4 (yellow), it can be changed to green

If 2 is changed it will contradict with 6 (blue), 4 (yellow), or 8 (green)

If 3 is changed it will contradict with 6, 5 (blue) or 8 (green), it can be changed to yellow

If 4 is changed it will contradict with 2, 7 (red) or 8 (green), it can be changed to blue

If 5 is changed it will contradict with 8 (green), 4 (yellow), or 1, 3 (red)

If 6 is changed it will contradict with 2, 3 (red) or 8 (green)

If 7 is changed it will contradict with 4 (yellow) or 8 (green)

If 8 is changed it will contradict with 2, 3, 7 (red), 5, 6 (blue), or 4 (yellow)

3. The graph has an Euler cycle according to the following path in vector notation:

Cycle (circuit):

1. (1,2)
2. (2,3)
3. (3,6)
4. (6,5)
5. (5,4)
6. (4,2)
7. (2,6)
8. (6,7)
9. (7,5)
10. (5,3)
11. (3,8)
12. (8,7)
13. (7,4)
14. (4,1)

All the vertices are of an even degree (0 at odd). Therefore, there must be at least one Euler cycle (circuit) in the graph.

The Eelier path is the Cycle shown above, as we learned, a cycle is a special occurrence of a path, meaning the above cycle is a path.

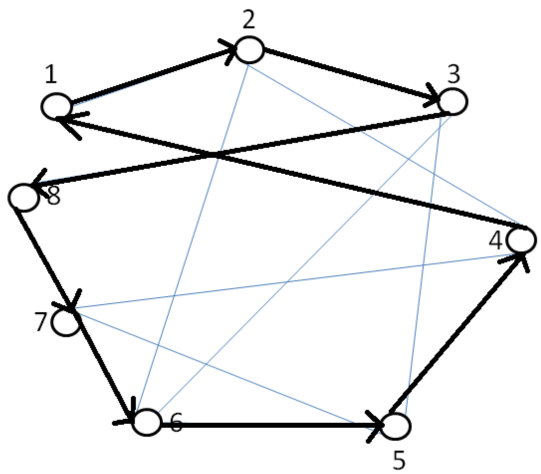
There is no Euler path that isn't also a cycle in this graph. In order for a graph to have an Euler path (that isn't a cycle), there must be 2 odd vertices, this graph has 0 odd vertices.
4.

The graph contains a Hamiltonian cycle, the following path for which is:

Start 8:

1. (8, 7)
2. (7, 6)
3. (6, 5)
4. (5, 4)
5. (4, 1)
6. (1, 2)
7. (2, 3)
8. (3, 8)

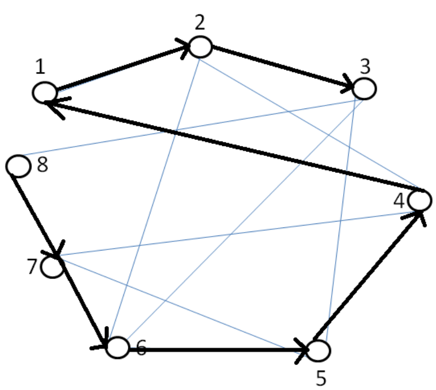
End 8:



Start and end are the same point, edges don't get trans-versed twice otherwise.

Technically, the above can be referenced as a Hamiltonian path with the removal of step 8.

1. (8, 7)
2. (7, 6)
3. (6, 5)
4. (5, 4)
5. (4, 1)
6. (1, 2)
7. (2, 3)



5. The following graphs are isomorphic.

The following nodes are equivalent:

1 = a

2 = c

3 = e

4 = b

5 = f

6 = d

The following edges are subsequently equivalent:

{a, c} = {1, 2}

{c, e} = {2, 3}

{e, b} = {3, 4}

{b, f} = {4, 5}

{f, d} = {5, 6}

{d, a} = {6, 1}

The vertices and edges of both graphs can be defined and lined up the same, the graphs are thus isomorphic, and the same number of vertices are connected the exact same way.

6.

a) $(9n - 4)^2$ is $\Theta(n^2)$

Prove O and Ω:

For O of n^2:

$81n^2 - \cancel{72n} + 16$, removed -72n

$< 81n^2 + 16n^2$, 16 is multiplied by factor n^2

$\leq 97n^2, \forall n \geq 1$

c = 97, k = 1

O of n^2, it's bigger when k=1, example:

Original: $81(1)^2 - 72(1) + 16 = 25$

For O of n^2: $97(1)^2 = 97$

True

For Ω of n^2:

$81n^2 - 72n + 16$

$> 81n^2 - 72n + \cancel{16}$, remove +16

$> 81n^2 - 72n^2$, multiply -72n by factor n

$> 9n^2, \forall n \geq 1$

$c = 9$, $k = 1$

Ω of n^2, it’s smaller when k=1, example:

Original: $81(1)^2 - 72(1) + 16 = 25$

For Ω of n^2: $9(1)^2 = 9$

True

Both O and Ω are true therefore,

$(9n - 4)^2$ is $\Theta(n^2)$ is true because O and Ω

True

b) For O of n^2

$8n^2 + n - 9$, math

$< 8n^2 + n – \cancel{9}$, removed - 9

$< 8n^2 + n^2$, multiplied n by factor of n

$\leq 9n^2, \forall n \geq 1$

$c = 9$, $k = 1$, O of n^2, it’s bigger when k=1, example:

Original: $8(1)^2 + (1) – 9 = 0$

For O: $9(1)^2 = 9$

True

c) For O of n^2:

$(20\log(n+7))/5$ is $O(n^2)$

$= 4 \log (n+7)$, simplification of division by math

$\leq 4 (n+7)$, removal of log by big O

$= 4n + 28$, simplification

$\leq 4n^2 + 28n^2$, multiplied everything by factors of n to n^2

$= 32n^2$

$c = 32$, $k = 1$, bigger when k=1, O of n^2, example: (using log base 2)

Original: (20log(1+7))/5 = 12

For O: $32(1)^2 = 32$

True

d) For Ω:

$5n^2 - 2n$ is $\Omega(1)$

$5n^2 - 2n \geq a * 1, \forall n \geq b$ for some $a > 0$ and $b > 0$

$5n^2 - 2n \geq 5 - 2 = 3 \forall n \geq 1$

Therefore, c = 3, k = 1, its true $5n^2 - 2n$ is $\Omega(1)$, its smaller or equal when k=1, example:

Original: $5(1)^2-2(1) = 3$

For Ω: 3

True

7. In order to determine which search is better they must be compared to each other:

Binary search: list is 32 index's long

1. 16 (value is smaller)
2. 8 (is smaller)
3. 4 (might be index 4, might be smaller)
4. 2 (if smaller then last step)
5. 1 OR 3 (must be one of these two, else it's not in index)

Binary search has a minimum (best case) of 3 searches, max (worst case) of 5.

Linear search: start from 1 (out of 32)

1. 1 (might be one, next otherwise)
2. 2 (next)
3. 3 (next)
4. 4 (last option)

Linear search has a minimum (best case) of 1 searches, max (worst case) of 4.

Linear searches average is (4+1)/2 = 2.5 searches, Binary average (3+5)/2 = 4 searches. Therefore the average amount of searches (between best and worst case scenarios) is better for linear searches.

Linear search also has a better best (best case1) and **worst case** scenario magnitude (worst case 4), meaning linear search is better by having a lower worst case scenario magnitude.

Linear search is the better option (assuming we start searching from index 1).

8. A) The worst case doubles in magnitude, as the number of indexes to be searched (under the worst case scenario) stretches by a factor of 2. If a list is 16 indexes long, and the quarry is the last item in the list, it will take 16 searches; double the list size, to 32 units, then the last quarry is the last item of 32 unit's magnitude, which takes 32 searches.

For example, in a list 4 units long, if the index searched is the last:

1. 1
2. 2
3. 3
4. 4

Worst case is 4 searches

If the list is doubled to 2n then:

1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8

The number of searches to the worst case doubles.

The worst case for a linear search increases by a multiple of n, if n is doubled, so is the number of searches.

B) The worst case scenario increases by a magnitude of 1 search, because binary searches halve each index in the list till they find the correct index. Doubling the list size creates 1 new search, to return to the original size (half the double list). Thus, it can be said a list the size of 2n indexes in a binary search will reach an index magnitude of n in one search, increasing the total magnitude of searches by 1 search if list size n is doubled.

For example, in a worst case scenario of 16 units, a binary search will search 4 times (for index 1 OR 3):

1. 8
2. 4
3. 2
4. 1 OR 3

When the list n is doubled to 2n then it is 32 indexes long:

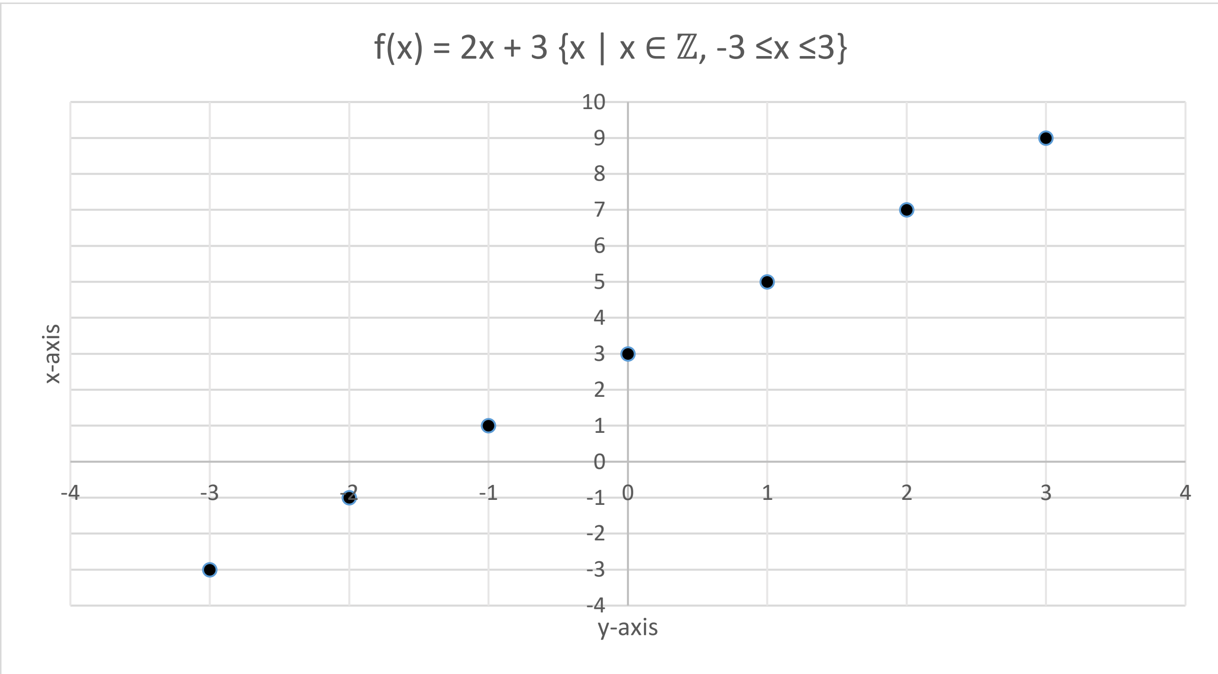
1. 16

2. 8
3. 4
4. 2
5. 1 OR 3

Thus it takes one more search.

The worst case for a binary search is creased by a sum of the multiple of the list. If the list is doubled, one more search is added to a binary search.

9.



10. ID: 101041125, a = 5, b = 2, c = 3a = 3*5 = 15, d = 2b = 2*2 = 4

f(x) = ax + c, z → z therefore, f(x) = 5x + 15 = 5(x+3)

g(x) = c^2 – d, z → z therefore, g(x) = 15^2 – 4 = 221

a) f o g = f(g(x)) = f(221) = 5(221+3) = 5(224) = 1120

b) g o f = g(f(x)) = g((5(x+3))) = g(5x + 15) = 221

c) (fog) o g = (f(g(x)) o g = f(g(g(x))) = f(g(221)) = f(221) = 5(221+3) = 1120