CSC165H1: Problem Set 4

Due December 6 before 10pm

Question1

(a)

Proof: (by contradiction)

Let $n \in \mathbb{N}$

Let G = (V, E) be an arbitrary graph with |V| = n

We know $\sum_{v \in V} d(v) = 2|E|$, which is an even number

Assume the number of vertices in G with odd degree is odd

We will prove this is wrong

Case 1: n is even

By our assumption the number of vertices in G with odd degree is odd

Then the sum of these degree should be an odd number, we call it sum1

Hence the number of vertices in G with even degree is odd

Then the sum of these degree should be an even number, we call it sum2

Then $\sum_{v \in V} d(v) = sum1 + sum2$, which is an odd number

But we know that $\sum_{v \in V} d(v)$ can only be an even number

So our assumption is wrong

Case 2: n is odd

By our assumption the number of vertices in G with odd degree is odd

Then the sum of these degree should be an odd number, we call it sum1

Hence the number of vertices in G with even degree is even

Then the sum of these degree should be an even number, we call it sum2

Then $\sum_{v \in V} d(v) = sum1 + sum2$, which is an odd number

But we know that $\sum_{v \in V} d(v)$ can only be an even number

So our assumption is wrong

Put all above together we know that our assumption must be wrong

This is the contradiction

Hence the number of vertices in G with odd degree must be even ■

(b)

$d(v_4) = 2$

Proof: We know that |V| = 4

Hence $d(v_4) \in \{0, 1, 2, 3\}$

①Since $d(v_3) = 3$, it must be adjacent to all three other vertices

So $d(v_4) \neq 0$

②If $d(v_4) = 1$

Since $d(v_3) = 3$, it must be adjacent to all three other vertices: v_1, v_2, v_4

We know that $d(v_1) = 1$ and $d(v_4) = 1$

Which means the only neighbor of them is both v_3

We know that $d(v_2) = 2$ and v_2, v_3 are adjacent

This means v_2 has another neighbor, v_1 or v_4

But we have already proved that the only neighbor of v_1 and v_4 is both v_3

This is a contradiction, so our assumption is wrong, $d(v_4) \neq 1$

③If $d(v_4) = 3$

Since $d(v_4) = 3$, it must be adjacent to all three other vertices: v_1, v_3, v_4

Since $d(v_3) = 3$, it must be adjacent to all three other vertices: v_1, v_2, v_4

So we find two neighbors of v_1: v_3 and v_4

But we already know that $d(v_1) = 1$ i.e. v_1 has only one neighbor

This is a contradiction, so our assumption is wrong, $d(v_4) \neq 3$

According to ①②③ we know that the only possible solution is that $d(v_4) = 2$ ■

(c)

Proof: Let $n \in \mathbb{N}^+$

Let G = (V, E) be an arbitrary graph with |V| = n

Let $v \in V$, v is an arbitrary vertex in G

Assume d(v)= n

Then by the definition of the degree

There are n vertices $v_1, v_2, \dots, v_n \in V$ s.t. $(v, v_i) \in E$ for i $\in \{1,2,\dots,n\}$

Since $v, v_1, v_2, \dots, v_n \in V$

We can conclude that $|V| \geq n + 1$ ■

(d)

Proof: Let $n \in \mathbb{N}$

Let G = (V, E) be an arbitrary graph with |V| = n

Assume $\forall v \in V, d(v) = 2$

WTS G has a cycle

Let $v_1, v_2, v_3, \dots, v_n \in V$

Pick out v_1

Since $d(v_1) = 2$

There must be another vertex adjacent to v_1, we call it v_2

Since $d(v_2) = 2$

There must be another vertex adjacent to v_2 except v_1, we call it v_3

If: v_3 is adjacent to v_1(since we know $d(v_1) = 2$)

Then v_1, v_2, v_3 is a cycle

If: v_3 is not adjacent to v_1

Since $d(v_3) = 2$

There must be another vertex adjacent to v_3 except v_2, we call it v_4

If: v_4 is adjacent to v_1(since we know $d(v_1) = 2$)

Then v_1, v_2, v_3, v_4 is a cycle

If: v_4 is not adjacent to v_1

Since $d(v_4) = 2$

There must be another vertex adjacent to v_3 except v_4, we call it v_5

……

At last, if all $v_3, v_4, v_4, \dots, v_{n-1}$ are not adjacent to v_1

Since $d(v_1) = 2$

v_n must adjacent to v_1

Then $v_1, v_2, v_3, \dots, v_n$ is a cycle

Put all above together we know that

If $\forall v \in V, d(v) = 2$ then G must has a cycle ■

(e)

Proof: Let $n \in \mathbb{N}^+$

Let G = (V, E) be an arbitrary graph with |V| = n

Assume $\forall v \in V, d(v) \geq |V| - 3$ and $|V| > 4$

Assume G is not connected, we will show that our assumption is wrong

Let $u, v \in V$ be a pair of disconnected vertices

Let x be an arbitrary integer and $\in [1, n-1]$

Let $G_1 = (V_1, E_1)$ obtained by picking u and x other arbitrary vertices from V and persisting the edges between them as in G

Let $G_2 = (V_2, E_2)$ obtained by picking v and n - x - 2 remaining vertices from V and persisting the edges between them as in G

Then $|V_1| = x + 1, |V_2| = n - x - 1$

By our assumption $\forall v \in V, d(v) \geq |V| - 3$

Hence $x \geq n - 3$ and $n - x - 2 \geq n - 3$

By assumption we also know that $|V| \geq 4$, which means that $n > 4$

Then we can get $x > 1$ and $x \leq 1$, which are contradict to each other

Hence our assumption that G is not connected is False

So we have proved that G must be connected ■

(f)

Let $n \in \mathbb{N}$

Let G = (V, E) be an arbitrary graph with |V| = n

①From 3(e) we know that:

For every graph G = (V, E), where $\forall v \in V, d(v) \geq |V| - 3$ and $|V| > 4$ is connected

$|V| \geq 4$ means $n > 4$

If $\forall v \in V, d(v) \geq |V| - 3$

Then we can say $d(v) \geq n - 3 > 1$

Since $\sum_{v \in V} d(v) = 2|E|$

We can say $|E| > \frac{n^2-3n}{2}$

So we can get the following statement:

$$\text{Let } n \in \mathbb{N}, \forall G = (V, E), \left(|V| = n \wedge (\forall v \in V, d(v) > 1) \wedge |E| > \frac{n^2-3n}{2} \wedge n > 4\right) \Longrightarrow$$

G is connected (1)

②From Example 6.6 and 6.7 we know that:

$$\text{Let } n \in \mathbb{N}, \forall G = (V, E), \left(|V| = n \wedge |E| \geq \frac{(n-1)(n-2)}{2} + 1\right) \Longrightarrow \text{G is connected (1)}$$

Connection: Both of these conditions are making some assumptions on vertices in G, to make sure that G is connected

Lack thereof: Statement (2) only makes assumption on |E|, namely, if |E| satisfies some conditions them G must be connected

However, statement (1) makes assumptions on all d(v), |E| and |V|, which makes Statement (1) more complex than (2), if any of these three assumptions is not satisfied, then the statement will be False and G will possibly still be disconnected

Question2

(a)

Proof: Let $n \in \mathbb{N}^+$

Let $p \in \mathbb{N}$

Let p be the smallest integer $\text{s.t.}\ 2^{p+1} > n$

Let $b_p, \dots, b_0 \in \{0,1\}$

WTS there is a unique representation of n in the following form: $n = \sum_{i=0}^{p} b_i 2^i$

We proof by contradiction

Assume there is more than one representation of n in the following form: $n = \sum_{i=0}^{p} b_i 2^i$

Namely let $a_p, \dots, a_0 \in \{0,1\}$

a_i and b_i are not all equal for $i \in \{0, \dots, p\}$ and $n = \sum_{i=0}^{p} a_i 2^i = \sum_{i=0}^{p} b_i 2^i$

Let $c \in \{0, \dots, p\}$

Let $a_i = b_i$ for $i \in \{c, \dots, p\}$

This means that $\sum_{i=c}^{p} a_i 2^i = \sum_{i=c}^{p} b_i 2^i$

If we want to show $\sum_{i=0}^{p} a_i 2^i = \sum_{i=0}^{p} b_i 2^i$

Then we need $a_0 2^0 + a_1 2^1 + \cdots + a_{c-1} 2^{c-1} = b_0 2^0 + b_1 2^1 + \cdots + b_{c-1} 2^{c-1}$

Namely $(a_0 - b_0)2^0 + (a_1 - b_1)2^1 + \cdots + (a_{c-1} b_{c-1})2^{c-1} = 0$

Notice that: $(a_0 - b_0)2^0 + (a_1 - b_1)2^1 + \cdots + (a_{c-1} b_{c-1})2^{c-1}$

is also in the form of a binary representation

Then we have to show: $(a_0 - b_0)2^0 + (a_1 - b_1)2^1 + \cdots + (a_{c-1} b_{c-1})2^{c-1}$

is a binary representation of 0

According to our definition of binary representation

We can find out that the only binary representation of 0 is $0 \cdot 2^0$

Which means that $a_0 - b_0 = 0$ and $c = 1$

However, by our definition about c, we know that $a_0 \neq b_0$

So $a_0 - b_0 = 0$ is impossible, this is the contradiction

So our assumption:

There is more than one representation of n in the form: $n = \sum_{i=0}^{p} b_i 2^i$ is wrong

Hence there is a unique representation of n in the following form: $n = \sum_{i=0}^{p} b_i 2^i$ ■

(b)

By 2(a) we know that for every number $n \in \mathbb{N}^+$, there is a unique representation of n in the following form: $n = \sum_{i=0}^{p} b_i 2^i$, where p is the smallest integer such that $2^{p+1} > n$, p is non-negative, and $b_p, \dots, b_0 \in \{0,1\}$.

So let us consider n = 0

When n = 0

Let p = 0 and b_0 = 0

Then $\sum_{i=0}^{p} b_i 2^i = b_0 2^0 = 0$ is the unique binary representation of n = 0

So we can conclude that every natural number n has a unique representation in the following form: $n = \sum_{i=0}^{p} b_i 2^i$, where $p \in \mathbb{N}$ depends on n and $b_p, \dots, b_0 \in \{0,1\}$.
The reason why it is impossible to make the domain of the previous proof in 2(a) "for every number $n \in \mathbb{N}$" is:
When n = 0, since 2^{p+1} will always greater than 0, there does not exists such a p which is non-negative and at the same time is the smallest integer such that $2^{p+1} > n$.
Namely, the previous statement in 2(a) will not be True when n = 0.

Question3

(a)

Proof: Let $n \in \mathbb{N}$

Given the set of inputs for Search: J_n, where for each input $(lst, x) \in J_n$, lst has length n, and x and the elements of lst are all between the numbers 1 and 10

Note that the there are 10 possible values of x, n elements in the lst and 10 possible values for each element in the lst

So $|J_n| = 10 \cdot 10^n$

We need to calculate $Avg_{search}(n) = \frac{1}{|J_n|} \sum_{(lst,x) \in J_n} \text{running time of search}(lst, x)$

①First we consider x = 1 and x is in the lst

We define S_n to be the set of all list has length n and the elements of lst are all between the numbers 1 and 10

Then firstly we calculate $\sum_{lst \in S_n} \text{running time of search}(lst, 1)$

The running time of search (lst, 1) is the number of loop iterations performed, and this Is exactly equal to the position that the first 1 appears in lst plus 1

Then it equals to $\sum_{lst \in S_n} \text{position of the first 1in lst plus 1}$

Then we can split up S_n based on the position that the first 1 appears

So it equals to $\sum_{i=0}^{n-1} \sum_{\substack{lst \in S_n \\ \text{first 1 is at lst[i]}}} \text{position of the first 1in lst plus 1}$

$$= \sum_{i=0}^{n-1} \sum_{\substack{lst \in S_n \\ \text{first 1 is at lst[i]}}} i + 1$$

Since there are $9^i 10^{n-i-1}$ such lists with the first 1 at lst[i]

So it equals to $= \sum_{i=0}^{n-1} (i+1) 9^i 10^{n-i-1}$

$$= \sum_{i=0}^{n-1} (i+1) 9^i \frac{1}{10^{i+1-n}}$$

$$= \sum_{i=0}^{n-1} (i+1) 9^i \frac{1}{10^i 10^{1-n}}$$

$$= \sum_{i=0}^{n-1} (i+1) 9^i \frac{1}{10^i} 10^{n-1}$$

$$= 10^{n-1} \sum_{i=0}^{n-1} (i+1) (\frac{9}{10})^i$$

$$= 10^{n-1} \sum_{i=0}^{n-1} i (\frac{9}{10})^i + (\frac{9}{10})^i$$

$$= 10^{n-1}\left(\sum_{i=0}^{n-1} i\left(\frac{9}{10}\right)^{i} + \sum_{i=0}^{n-1}\left(\frac{9}{10}\right)^{i}\right)$$

$$= 10^{n-1}\cdot\left(\frac{n\left(\frac{9}{10}\right)^{n}}{\frac{9}{10}-1} + \frac{\frac{9}{10}-\left(\frac{9}{10}\right)^{n+1}}{\left(\frac{9}{10}-1\right)^{2}} + \frac{\left(\frac{9}{10}\right)^{0}\left(1-\left(\frac{9}{10}\right)^{n}\right)}{1-\frac{9}{10}}\right)$$

$$= 10^{n-1}\cdot\left(-\frac{n\left(\frac{9}{10}\right)^{n}}{\frac{1}{10}} + \frac{\frac{9}{10}-\left(\frac{9}{10}\right)^{n+1}}{\frac{1}{100}} + \frac{1-\left(\frac{9}{10}\right)^{n}}{\frac{1}{10}}\right)$$

$$= 10^{n}\cdot\left(-n\left(\frac{9}{10}\right)^{n} + 10\cdot\frac{9}{10} - 10\cdot\left(\frac{9}{10}\right)^{n+1} + 1 - \left(\frac{9}{10}\right)^{n}\right)$$

$$= 10^{n}\cdot\left(10-(n+10)\left(\frac{9}{10}\right)^{n}\right)$$

②Now we consider x = 1 and x is not in the lst

$$\sum_{(lst,1)\in J_n} \text{running time of search}(lst,1) = n\cdot 9^{n}$$

Put ①② together,

We know that $RT_{search}(1) = n\cdot 9^{n} + 10^{n}\cdot\left(10-(n+10)\left(\frac{9}{10}\right)^{n}\right)$

Note that $x \in \{1,2,\ldots,10\}$

And in our above proof, we didn't really use any special properties of 1 at all, other Than the fact it was one of the numbers guaranteed to be in the list. So in fact, for any value of x between 1 and n, the same equality holds

So $\sum_{n=1}^{10} RT_{search}(n) = 10\left(n\cdot 9^{n} + 10^{n}\cdot\left(10-(n+10)\left(\frac{9}{10}\right)^{n}\right)\right)$

Hence $AVG_{search}(n) = \frac{1}{|J_n|}\sum_{n=1}^{10} RT_{search}(n)$

$$= \frac{10\left(n\cdot 9^{n}+10^{n}\cdot\left(10-(n+10)\left(\frac{9}{10}\right)^{n}\right)\right)}{10\cdot 10^{n}}$$

$$= \frac{n\cdot 9^{n}+10^{n}\cdot\left(10-(n+10)\left(\frac{9}{10}\right)^{n}\right)}{10^{n}}$$

$$= n\cdot\left(\frac{9}{10}\right)^{n} + 10 - (n+10)\left(\frac{9}{10}\right)^{n}$$

$$= 10 - 10\left(\frac{9}{10}\right)^{n}$$

$$= 10\left(1-\left(\frac{9}{10}\right)^{n}\right)$$

Since $n \in [1,10]$

$\left(\frac{9}{10}\right)^{n} \in (0,1)$

$10\left(1-\left(\frac{9}{10}\right)^{n}\right) \in (1,10)$

Hence $AVG_{search}(n)$ is $\Theta(1)$ ■

(b)

When the lst has length n and x and the elements of lst are all between the numbers 1 and 500

$\text{Avg}_{\text{search}}(n)$ is still $\Theta(1)$

Reason: The way to calculate this average run time is exactly the same as the way in 3(a),

Just replace 10 by 500 and replace 9 by 499

Hence the result is the same too, a constant with respect to the length of the input list, and is $\Theta(1)$

(c)

Proof: Let $s, u \in \mathbb{N}$ and u < 7

Define the input size of counter as n = s + u

We can find that the run time of if condition is 2 times as the else condition

In worst-case we need if condition runs as more as possible and else condition runs as less as possible

①First we prove $WC_{\text{counter}}(n) \in O(n)$

The if condition runs at most n iterations

The else condition runs at most 6n iterations

With each iteration taking constant time

So the while loop cost at most 7n steps

Ignore the first assignment step and the last return step

So the total cost of counter is at most 7n steps, which is O(n)

②Then we prove $WC_{\text{counter}}(n) \in \Omega(n)$

Consider the input family (n, 0) where s = n and u = 0

In this case, the if condition runs n iterations

The else condition runs 6n iterations

With each iteration taking constant time

So the while loop cost at most 7n steps

Ignore the first assignment step, and the last return step

So the total cost of counter is 7n steps, which is $\Omega(n)$

According to ①② we can conclude that $WC_{\text{counter}}(n) \in \Theta(n)$ ■

(d)

Proof: Let $s, u \in \mathbb{N}$ and u < 7

Define the input size of counter as n = s + u

We can find that the run time of if condition is 2 times as the else condition

In best-case we need if condition runs as less as possible and else condition runs as more as possible

①First we prove $BC_{\text{counter}}(n) \in O(n)$

The if condition runs at least n - 6 iterations

The else condition runs at least 6(n - 6) + 6 iterations

With each iteration taking constant time

So the while loop cost at least 7n - 36 steps

Ignore the first assignment step and the last return step

So the total cost of counter is at least 7n - 36 steps, which is O(n)

②Then we prove $BC_{\text{counter}}(n) \in \Omega(n)$

Consider the input family (n - u, 6), where s = n - u and u = 6

In this case, the if condition runs n - 6 iterations

The else condition runs 6(n - 6) iterations

With each iteration taking constant time

So the while loop cost 7n - 36 steps

Ignore the first assignment step and the last return step

So the total cost of counter is 7n - 36 steps, which is $\Omega(n)$

According to ①② we can conclude that $BC_{counter}(n) \in \Theta(n)$ ■

(e)

$AVG_{counter}(n) \in \Theta(n)$

Explain: By 3(c) we know that $WC_{counter}(n) \in \Theta(n)$

By 3(d) we know that $BC_{counter}(n) \in \Theta(n)$

Since the average-case is always between the best-case and the worst-case

We can conclude that $AVG_{counter}(n) \in \Theta(n)$