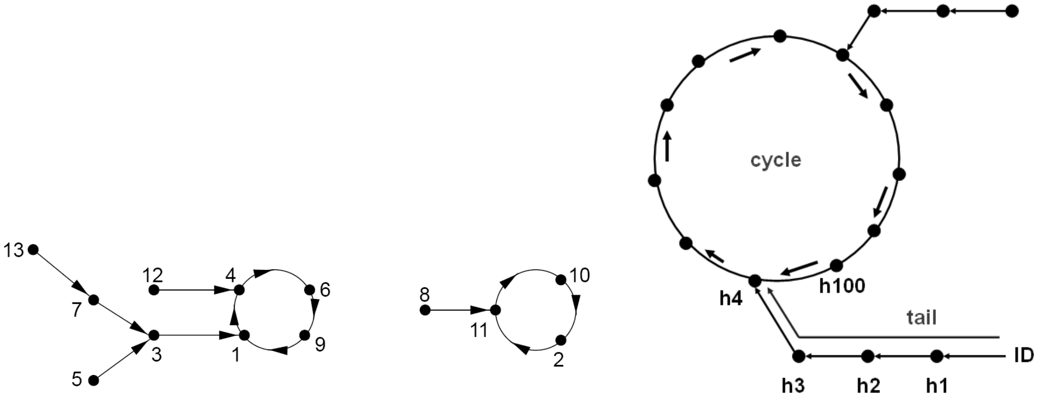
For a given hash function h, hash chain is recursively defined as follows:

$$H_0 = x$$
$$H_i = h(H_{i-1}) \text{ for } i \geq 1.$$

For the purpose of this homework, our hash function is the last k bits of MD5. In other words, $h|_k(x) =$ LSB k bits of $\mathrm{MD5}(x)$. After reading Section 2.1.6 of Handbook (available from `http://www.cacr.math.uwaterloo.ca/hac/about/chap2.pdf`), answer the following questions.

(a) (15 pts) Write a computer program to compute number of components, average/max tail length, min/average/max cycle length when we use $h|_{16}(\cdot)$. Your output should print out these 6 numbers. Tail and cycle are defined in 2.35. To avoid the confusion, *tail length* is defined as the number edges of the path to a cycle from a point. In the following Figure 2, the number of component is 2, tail length of node 13 is 3, tail length of node 12 is 1, and tail length of 6 is 0. Average tail length is (3+2+1+1)/4 = 1.75. (i.e. average of tail length starting from the terminal points that do not have the preimage.) The cycle length of [1, 4, 6, 9] is 4.



(a) (b)

Fig. 1. (a) Functional Graph in 2.1.6 and (b) Graph on Hash Collision

(b) (10 pts) How would you interpret the result in (a) in comparison with Fact 2.34 and 2.37 in HAC? What are the differences between these two?

(c) (10 pts) Under the assumption that MD5 is a random function, design a new hash chain that does not have cycle. Why your algorithm does not have cycle?

(d) (20 pts) Let $h|_k(\cdot)$ be the last k bits of MD5. Then, find a cycle of h. To show that you found a cycle, present the initial value, and the number of times it needs to be hashed before it repeats. Under the assumption that your program runs correctly, here's the grading criteria. If $k > 80$, you will get 20 points. If $72 < k \leq 80$, you will get 16 points. If $64 < k \leq 72$, you will get 12 points. If $56 < k \leq 64$, you will get 8 points. If $32 < k \leq 56$, you will get 4 points. For the purpose of verification, use your student ID as an initial hash value and provide the cycle length you found.

(e) (extra credit: 20 pts) This time, find a collision of h using k that you obtained in (d), that is, two distinct messages m_1, m_2 such that $h(m_1) = h(m_2)$. Grading scale is the same as (d). For the purpose of verification, use your student ID as an initial hash value and provide the following values: the collided hash (*h4* in the figure), and its preimages in the tail and the cycle (*h3* and *h100* in the figure below, respectively).
(Note: for (d) and (e), be careful not to use too much memory: a straightforward algorithm could use a few gigabytes. You need to use more clever, space-efficient algorithm. Either design one yourself, or do some research for such an algorithm.)