

DS HW5 Answer

Data Science HW5

1. Ans:

```
int h1(int key){
    int x = (key+7)*(key+7);
    x = x/16;
    x = x+key;
    x = x%11;
    return x;
}
```

M=11

key = 43: 代入 h1 $\Rightarrow 50 \times 50 = 2500 \Rightarrow 2500 / 16 = 156 \Rightarrow 156 + 43 = 199 \Rightarrow 199 \% 11 = 1$
 key = 23: 代入 h1 $\Rightarrow 30 \times 30 = 900 \Rightarrow 900 / 16 = 56 \Rightarrow 56 + 23 = 79 \Rightarrow 79 \% 11 = 2$

↓
 其他的key, 依此类推, 得出 "home slot"

key value	home slot	probe sequence
43	1	1
23	2	2
1	5	5
0	3	3
15	1	1, 2, 3, 4
31	0	0
4	0	0, 1, 2, 3, 4, 5, 6
7	8	8
11	9	9
3	9	9, 10

Final Hash Table
 ↓

Final Hash Table:

slot	0	1	2	3	4	5	6	7	8	9	10
content	31	43	23	0	15	1	4	7	11	3	

2.

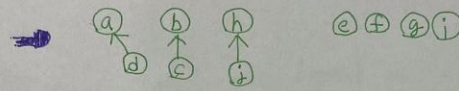
(a)

Ans:

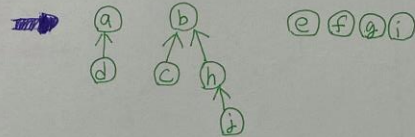
union(d,a)

union(b,c)

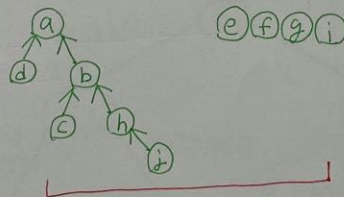
union(h,j)



union(h,b)



union(b,a)

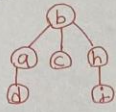


result up-tree forest

the depth of node j is 3#

(B) weight union

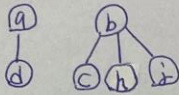
Ans:



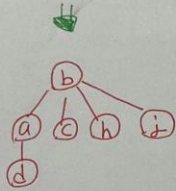
The depth of node j is 2_H

(C) with "weight union" and "path compression" after find(j)

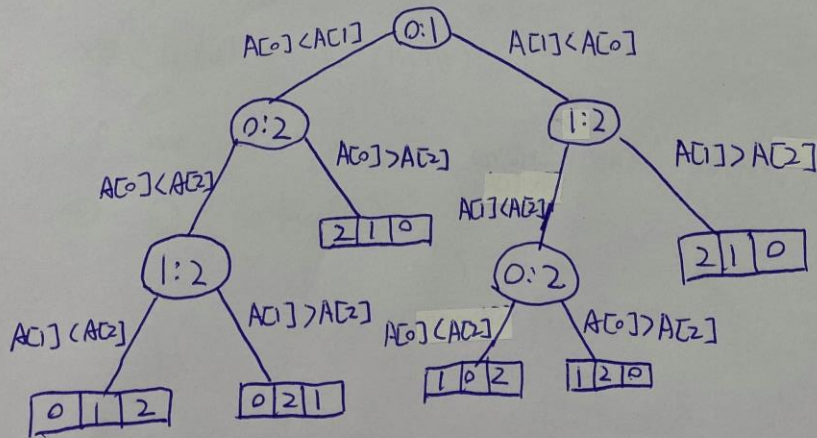
Ans:



The depth of node j is 1_H



3.



4.

(a) Count - Connected - Component (G):
count = n
for each vertex $v \in G, V$:
 makeset(v)
for each edge $(u, v) \in G, E$:
 if find-set(u) \neq find-set(v):
 Union(u, v)
 count = count - 1
return count.

After this procedure, the number of sets is the number of connected components.

(b) $O(n)$ for Make-set(v)
 $O(m)$ for find and union

~~\Rightarrow~~ Total = $O(n+m)$.

If we use "weighted union" and "path compression"

~~\Rightarrow~~ $O(m \cdot \log^* n)$

4.

(C)

Yes. The time complexity of (a) is $O(n+m)$.

The algorithm DFS and BFS can do it in $O(n+m)$ if using adjacency list.

→ 先任意從 graph 中, 挑選一個 vertex, 從它開始做 traversal, 直到全部皆被 visit ~~to~~ done.

→ 不論 breadth-first 或 depth-first traversal, time complexity 皆 $O(|V|+|E|)$