Assignment 7

ADITEYA BARAL [ab12057]

Q1 $m_0 = 23,\ m_1 = 17$

(a)

i	Operation $(m[i-2] \div m[i-1])$	Quotient	m_i
2	23/17	1	6
3	17/6	2	5
4	6/5	1	1
5	5/1	5	0

$$\Rightarrow \boxed{\begin{pmatrix} m \\ q \end{pmatrix} = \begin{pmatrix} 23 & 17 & 6 & 5 & 1 & 0 \\ - & 1 & 2 & 1 & 5 & - \end{pmatrix}}$$

(b) $s_0 = 1,\ s_1 = 0,\ s_{i+1} = s_{i-1} - q_i s_i;\ t_0 = 0,\ t_1 = 1,\ t_{i+1} = t_{i-1} - q_i t_i$

i	$s_{i+1} = s_{i-1} - q_i s_i$	$t_{i+1} = t_{i-1} - q_i t_i$
1	$s_2 = 1 - 1 \times 0 = 1$	$t_2 = 0 - 1 \times 1 = -1$
2	$s_3 = 0 - 2 \times 1 = -2$	$t_3 = 1 - 2 \times -1 = 3$
3	$s_4 = 1 - 1 \times -2 = 3$	$t_4 = -1 - 1 \times 3 = -4$
4	$s_5 = -2 - 5 \times 3 = -17$	$t_5 = 3 - 5 \times -4 = 23$

$$\Rightarrow \boxed{\begin{pmatrix} m \\ q \\ s \\ t \end{pmatrix} = \begin{pmatrix} 23 & 17 & 6 & 5 & 1 & 0 \\ - & 1 & 2 & 1 & 5 & - \\ 1 & 0 & 1 & -2 & 3 & 17 \\ 0 & 1 & -1 & 3 & -4 & 23 \end{pmatrix}}$$

(c) $s_k p + t_k a = \gcd(p,a) = 1$

$s_k(23) + t_k(17) = 1$

For $k=4 \rightarrow 3(23) + (-4)(17) = 1 \Rightarrow (-4)(17) \equiv 1 \pmod{23}$

$\Rightarrow \boxed{t_4 = a^{-1} = -4}$

Since we want an answer in $[0, 22]$, $\boxed{t_k = -4 + 23 = 19}$

$\Rightarrow \boxed{a^{-1} = 19}$

Verification $\rightarrow$

$17 \equiv -6 \pmod{23}$

$\Rightarrow -6(x) \equiv 1 \pmod{23}$

Let $x = -4 \rightarrow$

$-6(-4) = 24 \equiv 1 \pmod{23}$

Let $x = 19 \rightarrow$

$-6(19) = -114 \equiv 1 \pmod{23}$

$\Rightarrow$ $\boxed{\text{Inverse of } 17 \pmod{23} = -4 \text{ or } 19}$

Q2 (a) $a = (001'010) = x^4 + x^2$
$b = (101'101) = x^5 + x^3 + x^2 + 1$

Since $\mathbb{Z}_2$ is a ring with modulo 2, addition and subtraction are both bitwise XOR operations -

$$a+b = a-b = (001'010) \oplus (101'101)$$
$$= (100'111)$$

$$\Rightarrow \boxed{a+b = a-b = (100'111) = x^5 + x^4 + x^3 + 1}$$

(b) $a \cdot b = (x^4 + x^2) \cdot (x^5 + x^3 + x^2 + 1)$
$$= x^9 + x^6 + x^5 + x^2$$

Reducing using $(\text{mod } x^6 + x^5 + 1)$:

We know - $x^6 \equiv x^5 + 1 \pmod{f(x)}$

$$x^9 = x^3 \cdot x^6 \equiv x^3 (x^5 + 1) = x^8 + x^3$$
$$x^8 = x^2 \cdot x^6 \equiv x^2(x^5+1) = x^7 + x^2$$
$$x^7 = x \cdot x^6 \equiv x(x^5+1) = x^6 + x = x^5 + 1 + x$$

$$\Rightarrow a \cdot b \equiv x^8 + x^3 + x^6 + x^5 + x^2$$
$$\equiv x^7 + x^2 + x^3 + x^6 + x^5 + x^2$$
$$\equiv x^5 + 1 + x + x^2 + x^3 + x^5 + 1 + x^5 + x^2$$
$$\equiv 3x^5 + x^3 + 2x^2 + x + 2$$
$$\equiv x^5 + x^3 + x \pmod{f(x)}$$

$$\Rightarrow \boxed{a \cdot b = (010'101) = x^5 + x^3 + x}$$

(c) $m_0 = x^6 + x^5 + 1$, $m_1 = x^5 + x^3 + x^2 + 1$

i	Quotient $(m[i-2] / m[i-1])$	m_i
2	$x+1$	$x^4 + x^2 + x$
3	x	1

$\gcd(f(x), g(x)) = 1$.

$$\boxed{\begin{pmatrix} m \\ q \end{pmatrix} = \begin{pmatrix} x^6+x^5+1 & x^5+x^3+x^2+1 & x^4+x^2+x & 1 \\ - & x+1 & x & - \end{pmatrix}}$$

(d) $s_0 = 1$, $t_0 = 0$
$s_1 = 0$, $t_1 = 1$

i	Quotient $(m[i-2] / m[i-1])$	m_i	s_i	t_i
2	$x+1$	$x^4 + x^2 + x$	1	$x+1$
3	x	1	x	$x^2 + x + 1$

$$\boxed{\begin{pmatrix} m \\ q \\ s \\ t \end{pmatrix} = \begin{pmatrix} x^6+x^5+1 & x^5+x^3+x^2+1 & x^4+x^2+x & 1 \\ - & x+1 & x & - \\ 1 & 0 & 1 & x \\ 0 & 1 & x+1 & x^2+x+1 \end{pmatrix}}$$

(e) From (d),

$$x f(x) + (x^2 + x + 1) g(x) = 1$$

$$\Rightarrow g^{-1}(x) = t_3(x)$$
$$= x^2 + x + 1$$

$$\Rightarrow \boxed{g^{-1}(x) = x^2 + x + 1 = (111'000)}$$

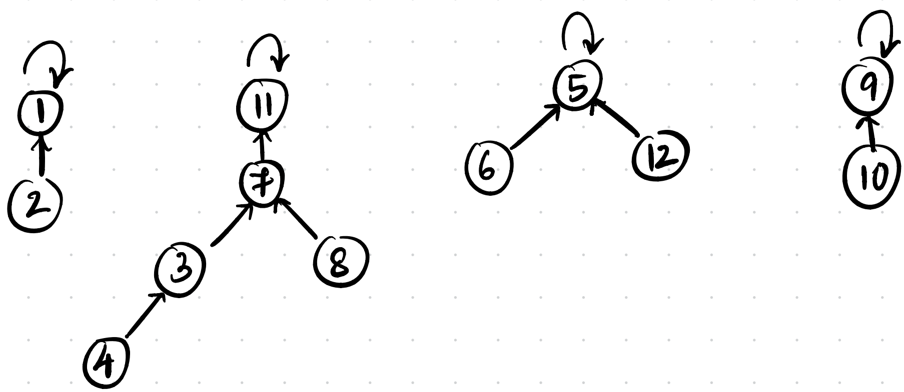
Q3 $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$

$\pi[i] \leftarrow i$, $\pi[2i] \leftarrow 2i-1$

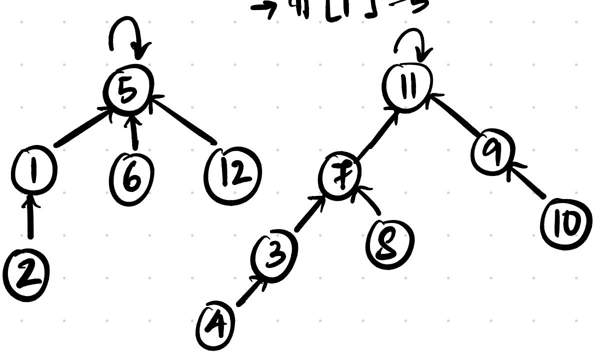
$\Rightarrow U = \{ \{\underline{1}, 2\}, \{\underline{3}, 4\}, \{\underline{5}, 6\}, \{\underline{7}, 8\}, \{\underline{9}, 10\}, \{\underline{11}\}, \{\underline{12}\} \}$

(a) $\xrightarrow{\text{Union}(4,8)}$ $\pi[4] = 3$, $\pi[8] = 7$, $\rightarrow \pi[3] = 7$ $\xrightarrow{\text{Union}(12,6)}$ $\pi[12] = 12$, $\pi[6] = 5$, $\rightarrow \pi[12] = 5$ $\xrightarrow{\text{Union}(3,11)}$ $\pi[3] = 7$, $\pi[11] = 11$, $\rightarrow \pi[7] = 11$

$\xrightarrow{\text{Find}(8)}$ $\boxed{11}$



Union $(2,6) \rightarrow$ $\pi(2) = 1$, $\pi(6) = 5$, $\rightarrow \pi[1] = 5$ $\xrightarrow{\text{Union}(10,11)}$ $\pi[10] = 9$, $\pi[11] = 11$, $\rightarrow \pi[9] = 11$ $\xrightarrow{\text{Find}(4)}$ $\boxed{11}$



(b) $U = \{\{\underline{1},2\}, \{\underline{3},4\}, \{\underline{5},6\}, \{\underline{7},8\}, \{\underline{9},10\}, \{\underline{11}\}, \{\underline{12}\}\}$

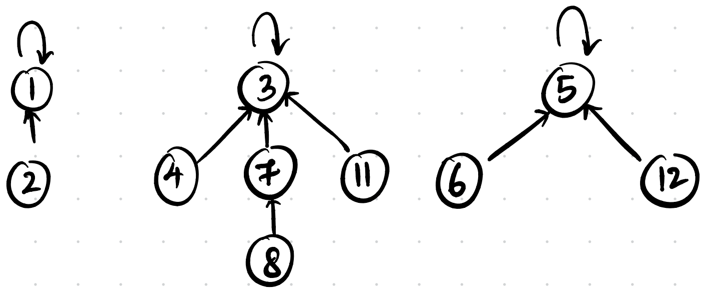
size = $\{2,1,2,1,2,1,2,1,2,1,1,1\}$

$\xrightarrow{\text{Union}(4,8)}$ $\pi[4]=3$
$\pi[8]=7$
$\rightarrow \pi[7]=3$
$s=\{2,1,4,1,2,1,2,1,2,1,1,1\}$

$\xrightarrow{\text{Union}(12,6)}$ $\pi[12]=12$
$\pi[6]=5$
$\rightarrow \pi[12]=5$
$s=\{2,1,4,1,3,1,2,1,2,1,1\}$

$\xrightarrow{\text{Union}(3,11)}$ $\pi[3]=3$
$\pi[11]=11$
$\rightarrow \pi[11]=3$
$s=\{2,1,5,1,3,1,2,1,2,1,1\}$

$\xrightarrow{\text{Find}(8)}$ $\boxed{3}$

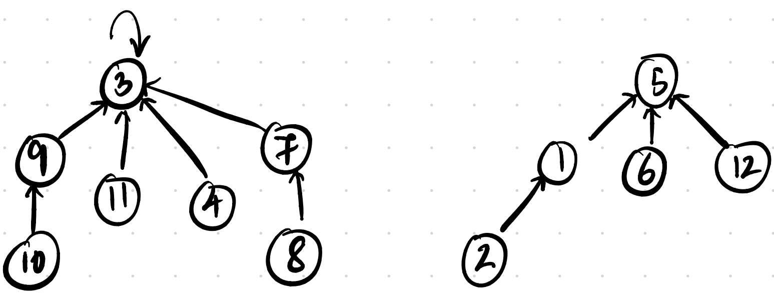


size = $\{2,1,5,1,3,1,2,1,2,1,1\}$

$\xrightarrow{\text{Union}(2,6)}$ $\pi[2]=1$
$\pi[6]=5$
$\rightarrow \pi[1]=5$
$s=\{2,1,5,1,5,1,2,1,2,1,1\}$

$\xrightarrow{\text{Union}(10,11)}$ $\pi[10]=9$
$\pi[11]=3$
$\rightarrow \pi[9]=3$
$s=\{2,1,7,1,5,1,2,1,2,1,1\}$

$\xrightarrow{\text{Find}(4)}$ $\boxed{3}$

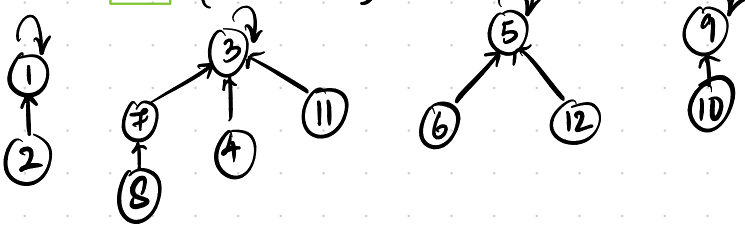


(c) $\xrightarrow{\text{Union }(4,8)}$ $\pi[4] = 3$
$\pi[8] = 7$
$\pi[7] = 3$
$r = \{1,0,2,0,1,0,$
$1,0,1,0,0,0\}$

$\xrightarrow{\text{Union }(12,6)}$ $\pi[12] = 12$
$\pi[6] = 5$
$\pi[12] = 5$
$r = \{1,0,2,0,2,$
$0,1,0,1,0,0,0\}$

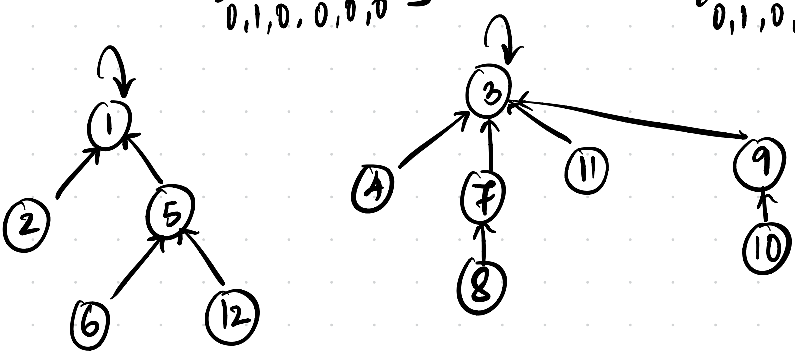
$\xrightarrow{\text{Union }(3,11)}$ $\pi[3] = 3$
$\pi[11] = 11$
$\pi[11] = 3$
$r = \{1,0,3,0,2,0,$
$1,0,1,0,0,0\}$

$\xrightarrow{\text{Find }(8)}$ $\boxed{3}$ (rank = 0)



$\xrightarrow{\text{Union }(2,6)}$ $\pi[2] = 1$
$\pi[6] = 5$
$\pi[5] = 1$
$r = \{2,0,3,0,2,0,1,$
$0,1,0,0,0,0\}$

$\xrightarrow{\text{Union }(10,11)}$ $\pi[10] = 9$
$\pi[11] = 3$
$\pi[9] = 3$
$r = \{2,0,3,0,2,0,1,$
$0,1,0,0,0,0\}$



$\xrightarrow{\text{Find }(4)}$ $\boxed{3}$

(d) $\xrightarrow{\text{Union}(4,8)}$

$\pi[4] = 3$
$\pi[8] = 7$
$\pi[3] = 7$

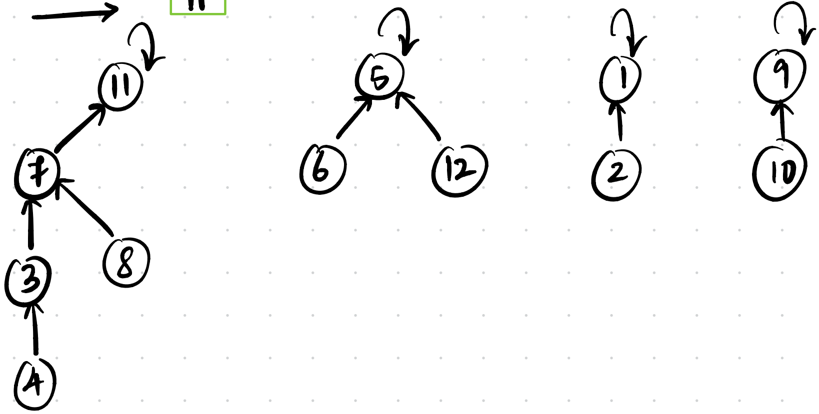
$\xrightarrow{\text{Union}(12,6)}$

$\pi[12] = 12$
$\pi[6] = 5$
$\pi[12] = 5$

$\xrightarrow{\text{Union}(3,11)}$

$\pi[3] = 7$
$\pi[11] = 11$
$\pi[7] = 11$

$\xrightarrow{\text{Find}(8)}$ $\boxed{11}$



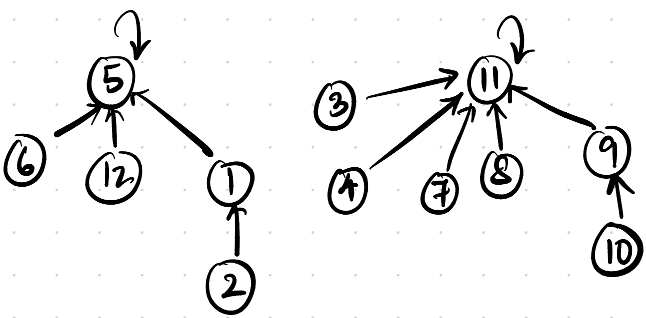
$\xrightarrow{\text{Union}(2,6)}$

$\pi[2] = 1$
$\pi[6] = 5$
$\pi[1] = 5$

$\xrightarrow{\text{Union}(10,11)}$

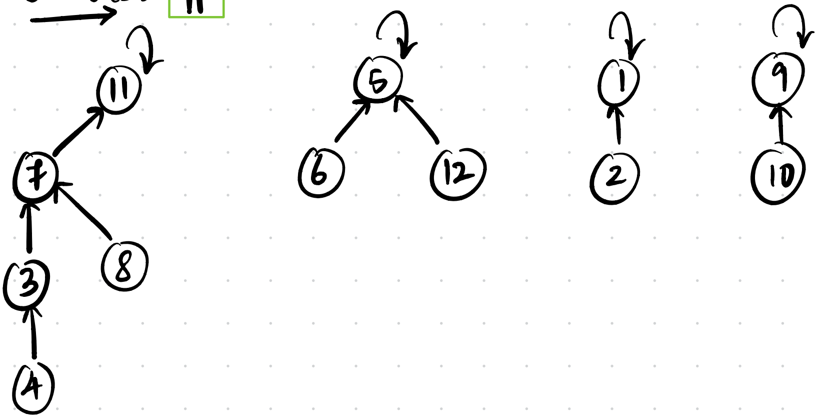
$\pi[10] = 9$
$\pi[11] = 11$
$\pi[9] = 11$

$\xrightarrow{\text{Find}(4)}$ $\boxed{11}$



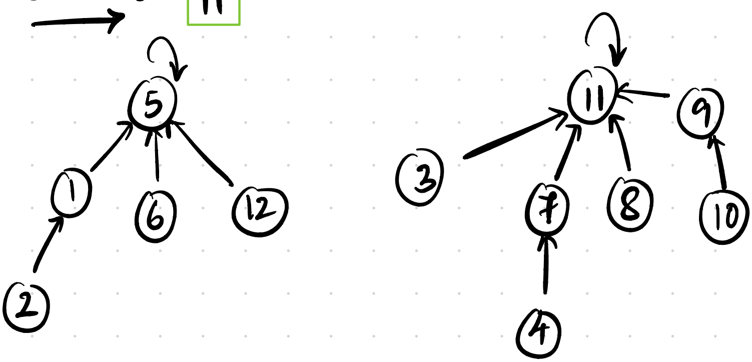
(e) $\xrightarrow{\text{Union}(4,8)}$ $\pi[4]=3$, $\pi[8]=7$, $\pi[3]=7$ $\xrightarrow{\text{Union}(12,6)}$ $\pi[12]=12$, $\pi[6]=5$, $\pi[12]=5$ $\xrightarrow{\text{Union}(3,11)}$ $\pi[3]=7$, $\pi[11]=11$, $\pi[7]=11$

$\xrightarrow{\text{Find}(8)}$ $\boxed{11}$



$\xrightarrow{\text{Union}(2,6)}$ $\pi[2]=1$, $\pi[6]=5$, $\pi[1]=5$ $\xrightarrow{\text{Union}(10,11)}$ $\pi[10]=9$, $\pi[11]=11$, $\pi[9]=11$

$\xrightarrow{\text{Find}(4)}$ $\boxed{11}$

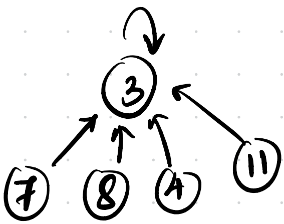


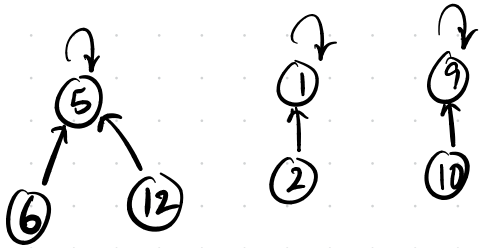
(f) $\xrightarrow{\text{Union}(4,8)}$ $\pi[4]=3$
$\pi[8]=7$
$\pi[7]=3$
$r=\{1,0,2,0,1,0,1,0,1,0,0,0\}$

$\xrightarrow{\text{Union}(12,6)}$ $\pi[12]=12$
$\pi[6]=5$
$\pi[12]=5$
$r=\{1,0,2,0,2,0,1,0,1,0,0,0\}$

$\xrightarrow{\text{Union}(3,11)}$ $\pi[3]=3$
$\pi[4]=11$
$\pi[11]=3$
$r=\{1,0,2,0,2,0,1,0,1,0,0,0\}$

$\xrightarrow{\text{Find}(8)}$ $\boxed{3}$

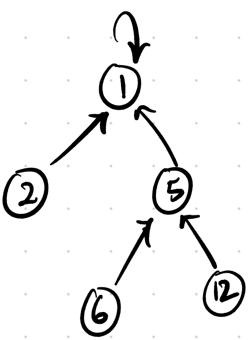


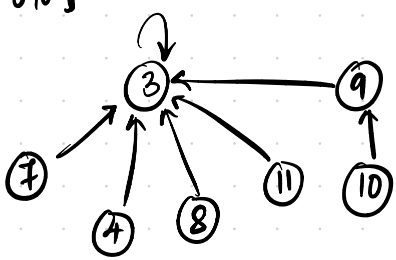


$\xrightarrow{\text{Union}(2,6)}$ $\pi[2]=1$
$\pi[6]=5$
$\pi[5]=1$
$r=\{2,0,2,0,2,0,1,0,1,0,0,0\}$

$\xrightarrow{\text{Union}(10,11)}$ $\pi[10]=9$
$\pi[11]=3$
$\pi[9]=3$
$r=\{2,0,5,0,2,0,0,0,1,0,0,0\}$

$\xrightarrow{\text{Find}(4)}$ $\boxed{3}$





Q4 (a) The Find operation always takes $O(1)$ constant time when all trees maintain a fixed depth of 2. This also reduces the number of traversals in Union from n to $\sqrt{n}$.

The sibling pointers help with quicker reassignment of the subtree using recursion. The child pointers are used to get the non-full internal node and corresponding leaf children. Having a circular sibling list allows $O(1)$ pointer redirection operations when merging an entire bucket under a new parent. The child pointer also allows arbitrary access to the circular list of siblings and allows tracking full and non-dull children.

Leaves are between 1 and $2 \log n$ to ensure time complexity remains $\log(\log n)$ such that no internal node is too heavy.

(b) NODE:

id
parent
child
sibling
degree
is root

Find (x):
Return x. parent. parent if x. parent else x

```
Union (x, y):

    if (x.degree < y.degree)
        swap (x, y)

    x' ← getNonFull (x)
    y' ← getNonFull(y)

    if (x'.degree >= y'.degree)
        Link (y', x')

    else
        Link (x', y')
        y'.parent ← x'
        x'.degree ← x'.degree + 1
        Discard (y', y)

GetNonFull (x):
    child ← x.child
    curr ← child
    while True
        if curr.degree < log n
            return curr
        curr ← curr.sibling


Link (x, y):
  child ← x.child
  children ← {child}
  while True
      child ← child.sibling
      if child is not φ
          children +← {child}
      else
          break
```

for each child in children
    child . parent $\leftarrow$ y
    y. degree $\leftarrow$ x. degree + 1

x. child $\leftarrow \phi$

(c) Each call for Find takes $O(1)$ time.
$\Rightarrow$ For m operations, total cost for Find = m

Each full node after merging needs to satisfy

$x1.\text{degree} + x2.\text{degree} \leq 2 \log N$

Thus, we can merge in $O(\log \log N)$ time

Number of full nodes = $\frac{N}{\log N}$

$\Rightarrow$ Total cost = $O\left(\frac{N}{\log N} \times \log(\log N)\right)$

Each union takes $O(1)$ time
For a total of m maximum union operations, cost = $O(m)$

Charges to items = $O(n)$

$$\boxed{\Rightarrow \text{Total cost} = O\left(m + m + n + \frac{n \log \log n}{\log n}\right) \approx O(m + n \log \log n)}$$

Q5 In a min-path DAG, we have all vertices in V that are reachable from s and only those edges that are part of a minimum cost path from s to $r \in V$.

The min path tree from s will have $|V|-1$ edges. However, since multiple edges can be part of min paths from the same source s to destination, the min path DAG can include more edges than a simple tree. This means that it has more than $|V|-1$ edges. Worst case, we have $O(|E|)$ edges and can have $O(|V|^2)$ size.

Consider a family of graphs where,

$V = \{1, 2, \dots, n\}$, $s = 1$.
For every pair $1 \le i < j \le n$, we have edge $i \to j$
We assign $C(i \to j) = 1$ for $\forall i < j$. Since all edges have positive cost, any path from 1 to j has cost equal to its length.

For each $j > 1$, any one-edge path has cost $= 1$ and any path $1 \to i \to \dots \to j$ has cost ≥ 2. $\Rightarrow$ Every direct edge $1 \to j$ is the shortest path from 1 to j. For each $j > 1$, every edge $(1 \to j)$ appears in the min path DAG. Now, if we make all edges $i \to j$ to be equally cheap and satisfy the shortest path property, $[C(i \to j) = 1 \ \forall i < j]$, then every edge $(i \to j)$ with $i < j$ is itself a min-cost path from $1 \to j$. Thus, the min-cost path contains all edges.

$\Rightarrow$ $\boxed{\text{Worst case, it is } O(|V|^2)}$

Q6

(a)

Stage	A	B	C	D	E	F	G	H
0	0	∞	∞	∞	∞	∞	∞	∞
1	0	1	5	11				
2		1	3		11			
3			3		9	4		
4				9	8	4	7	5
5					7		6	5
6				7			6	
7				7				
8					7			

S0: $d[A] \leftarrow 0$, $d[B \ldots H] \leftarrow \infty$

S1: Node $\leftarrow$ A

S2: Node $\leftarrow$ B

$$\left.\begin{array}{ll} B \to C & [C=3] \\ B \to E & [E=11] \end{array}\right\} \text{ 2 keys}$$

S3: Node $\leftarrow$ C

$$\left.\begin{array}{ll} C \to E & [E=4] \\ C \to F & [F=4] \end{array}\right\} \text{ 2 keys}$$

S4: Node ← F

$$\left.\begin{array}{l} F \to D \ [D=9] \\ F \to E \ [E=8] \\ F \to G \ [G=7] \\ F \to H \ [H=5] \end{array}\right\} 4 \text{ keys}$$

S5: Node ← H

$$\left.\begin{array}{l} H \to E \ [E=7] \\ H \to G \ [G=6] \end{array}\right\} 2 \text{ keys}$$

S6: Node ← G

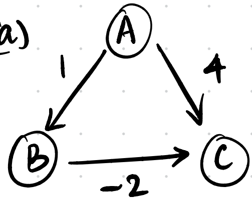
$$\left.G \to D \ [D=7]\right\} 1 \text{ key}$$

S7: Node ← D } 0 keys

S8: Node ← E } 0 keys

(b) $\boxed{\text{Total number of decreaseKey operations} = 11}$

Q7 (a)



Shortest paths from A:

$A \to A = 0$
$A \to B = 1$
$A \to C$ (via B) $= 1 - 2 = -1$

However, Dijkstra gives the following distances from A:

$A \to A = 0$
$A \to B = 1$
$A \to C = 4$.

This occurs because when we visit B, it does not update C since it assumes it is best visited through A. Thus, it takes cost $A \to C$ as 4 which is not the min cost.

(b)
```
Modified Djikstra (G, s)
    for ∀ v ∈ V
        d[v] ← ∞
    d[s] ← 0
    Q. insert (s, d[s])

    while Q != φ
        u = Q.deleteMin ()
        for ∀ v such that v is adjacent to u
            if d[v] > d[u] + cost (u,v)
                d[v] = d[u] + cost (u,v)
                if v in Q
                    Q.decreaseKey (v, d[v])
```

else

Q.insert (v, d[v])

Proof:

A non-cyclic path in a graph with n vertices can have at most (n-1) edges. A new non-cyclic path with lower cost is found every time d[v] is updated. Each vertex can be inserted into queue Q when we find a shorter path to it. Since there are only finitely only non-cyclic paths ⇒ each vertex can be inserted a finite number of times.

⇒ The algorithm will always terminate.

Q8 (a)

$k=a$

	a	b	c	x	y	z
a	0	4		2	5	
b		0	2		2	6
c			0	1		
x				0	-3	
y					0	-1
z			2			0

$k=b$

	a	b	c	x	y	z
a	0	4	6	2	5	10
b		0	2		2	6
c			0	1		
x				0	-3	
y					0	-1
z			2			0

k=c

	a	b	c	x	y	z
a	0	4	6	2	5	10
b		0	2	3	2	6
c			0	1		
x				0	-3	
y					0	-1
z			2	3		0

k=x

	a	b	c	x	y	z
a	0	4	6	2	-1	10
b		0	2	3	0	6
c			0	1	-2	
x				0	-3	
y					0	-1
z			2	3	0	0

$k=y$

	a	b	c	x	y	z
a	0	4	6	2	-1	-2
b		0	2	3	0	-1
c			-1	1	-2	-3
x				0	-3	-4
y					0	-1
z			2	3	0	-1

$k=z$

	a	b	c	x	y	z
a	0	4	0	1	-2	-3
b		0	1	2	-1	-2
c			-1	0	-3	-4
x			-2	-1	-4	-5
y			1	2	-1	-2
z			1	2	-1	-2

(b)

	a	b	c	x	y	z
a	0	4		2	5	
b		0	2		2	6
c			0	1		
x				0	-3	
y					0	-1
z						0

w/o intermediates & intermediates = {a} (1) & intermediates = {a, b} (2)

	a	b	c	x	y	z
a	0	4	6	2	-1	-2
b		0	2	3	0	-1
c			0	1	-2	-3
x				0	-3	-4
y					0	-1
z						0

intermediates = {a, b, c} (3) & = {a, b, c, x} (4) intermediates = {a, b, c, x, y} 5

	a	b	c	x	y	z
a	0	4	6	2	-1	-2
b		0	2	3	-3	-4
c			0	1	-2	-3
x				0	-3	-4
y					0	-1
z						0

intermediates = {a, b, c, x, y, z} 6

	a	b	c	x	y	z
a	0	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$
b		0	$-\infty$	$-\infty$	$-\infty$	$-\infty$
c			0	$-\infty$	$-\infty$	$-\infty$
x				0	$-\infty$	$-\infty$
y					0	$-\infty$
z						0

$\rightarrow$ $\boxed{\text{Total matrices} = 6}$