

### Solutions to Assignment 10

Given a sorted array and  $x$ , return  $i$ , such that  $\text{arr}[i] \leq x < \text{arr}[i+1]$ . If  $x < \text{arr}[0]$ , return  $-1$ . If  $x \geq \text{arr}[n-1]$ , return  $n-1$ .

int binarySearch(int[] arr, x):

```

n ← arr.length
if x < arr[0] then return -1
else if x ≥ arr[n-1] then return n-1
else return binarySearch(arr, 0, n-1, x)

```

// Pre:  $\text{arr}[\text{left}] \leq x < \text{arr}[\text{right}]$

binarySearch ( arr, left, right, x ):

n ← right - left + 1

if n = 2 then

return left

else

mid ← ( left + right ) / 2

if x < arr [ mid ] then

return binarySearch ( arr, left, mid, x)

else /\* arr[mid] ≤ x \*/

return binarySearch ( arr, mid, right, x)

### Solutions to Assignment 11

Prim's algorithm on the given example. Edges added to PQ in each step is shown in blue. Note that in some steps, a different edge with the same weight could have been removed from PQ and the path to the solution changes.

S	PQ
$S = \{7\}$	(7,6):3 (7,4):9 (7,5):3
Add (7,6) to mst: $S = \{7,6\}$ . wbst: 3	(7,4):9 (7,5):3 (6,3):4 (6,4):1
Add (6,4) to mst: $S = \{7,6,4\}$ . wbst: 4	(7,4):9 (7,5):3 (6,3):4 (4,3):2 (4,1):5 (4,2):4 (4,5):2
Add (4,3) to mst: $S = \{7,6,4,3\}$ . wbst: 6	(7,4):9 (7,5):3 (6,3):4 (4,1):5 (4,2):4 (4,5):2 (3,1):6
Add (4,5) to mst: $S = \{7,6,4,3,5\}$ . wbst: 8	(7,4):9 (7,5):3 (6,3):4 (4,1):5 (4,2):4 (3,1):6 (5,2):3
Discard (7,5), Add (5,2) to mst: $S = \{7,6,4,3,5,2\}$ . wbst: 11	(7,4):9 (6,3):4 (4,1):5 (4,2):4 (3,1):6 (2,1):4
Discard (6,3), (4,2), Add (2,1) to mst: $S = \{7,6,4,3,5,2,1\}$ . wbst: 15	(7,4):9 (4,1):5 (3,1):6
Discard (4,1), (3,1), (7,4).	q is empty - done

⑦

⑦ — 3 — ⑥

⑦ — 3 — ⑥ — ④

⑦ — 3 — ⑥ — 1 — ④ — 2 — ③

⑦ — 3 — ⑥ — 1 — ④ — 2 — ③  
④ — 2 — ⑤

⑦ — 3 — ⑥ — 1 — ④ — 2 — ③  
④ — 2 — ⑤  
② — 3 — ⑤

⑦ — 3 — ⑥ — 1 — ④ — 2 — ③  
④ — 2 — ⑤  
① — 4 — ② — 3 — ⑤

MST

Kruskal's algorithm on the given example:

After makeSet:

	1	2	3	4	5	6	7
parent	1	2	3	4	5	6	7
rank	0	0	0	0	0	0	0

find(4)=4, find(6)=6,

add (4,6) to mst, wmst: 1, union(4,6):

	1	2	3	4	5	6	7
parent	1	2	3	4	5	4	7
rank	0	0	0	1	0	0	0

find(3)=3, find(4)=4,

add (3,4) to mst, wmst: 3, union(3,4):

	1	2	3	4	5	6	7
parent	1	2	4	4	5	4	7
rank	0	0	0	1	0	0	0

find(4)=4, find(5)=5,

add (4,5) to mst, wmst: 5, union(4,5):

	1	2	3	4	5	6	7
parent	1	2	4	4	4	4	7
rank	0	0	0	1	0	0	0

find(5)=4, find(2)=2,

add (5,2) to mst, wmst: 8, union(4,2):

	1	2	3	4	5	6	7
parent	1	4	4	4	4	4	7
rank	0	0	0	1	0	0	0

find(6)=4, find(7)=7,

add (6,7) to mst, wmst: 11, union(4,7):

	1	2	3	4	5	6	7
parent	1	4	3	4	4	4	4
rank	0	0	0	1	0	0	0

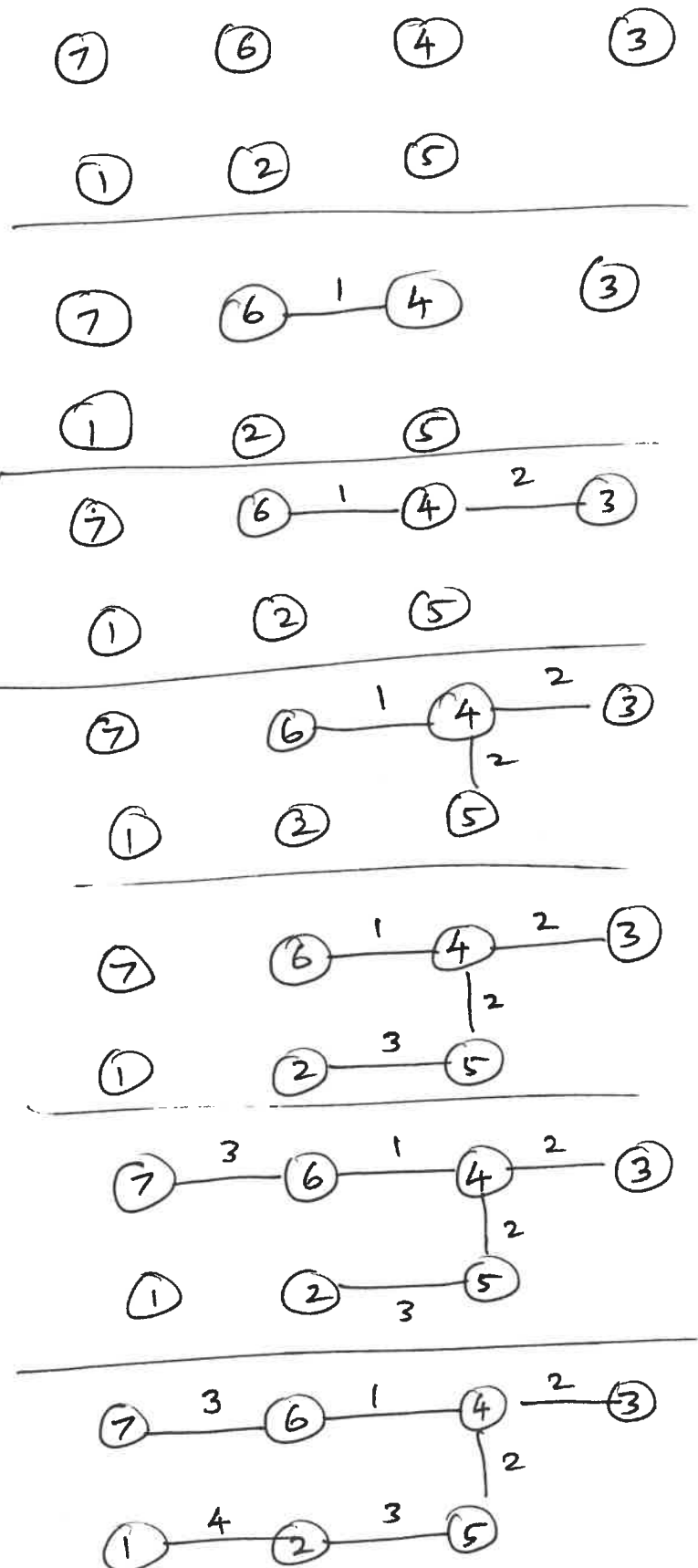
find(5)=4, find(7)=4, discard (5,7).

find(1)=1, find(2)=4,

add (1,2) to mst, wmst: 15, union(1,4):

	1	2	3	4	5	6	7
parent	4	4	3	4	4	4	4
rank	0	0	0	1	0	0	0

Remaining edges are discarded.



MST