Binomial Heaps

Abstract Data Types

Abstract Data Types		Insert	Min	Extract_Min	Union
Mergeable Priority Queues		✓	✓	✓	✓

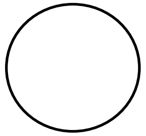
Data Structures

Abstract Data Types	Data Structures	Insert	Min	Extract_Min	Union
Mergeable Priority Queues	**Min Binomial Heap**	O(log n)	O(log n)	O(log n)	O(log n)

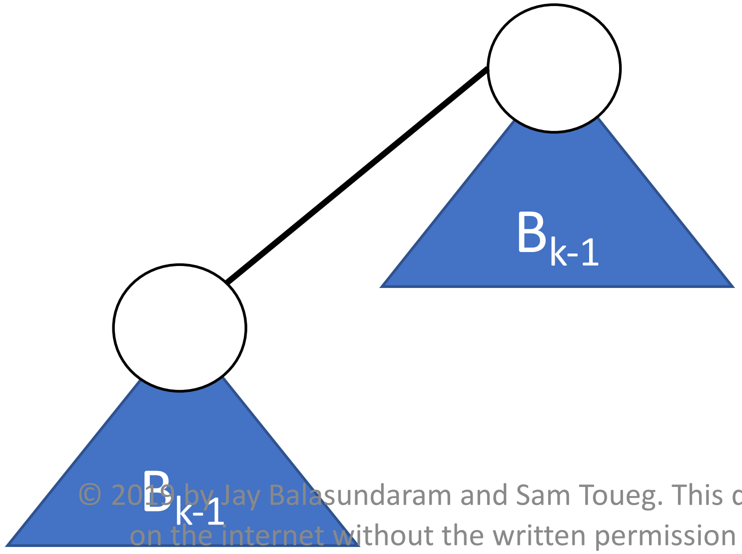
Binomial Trees

B_k tree: defined recursively

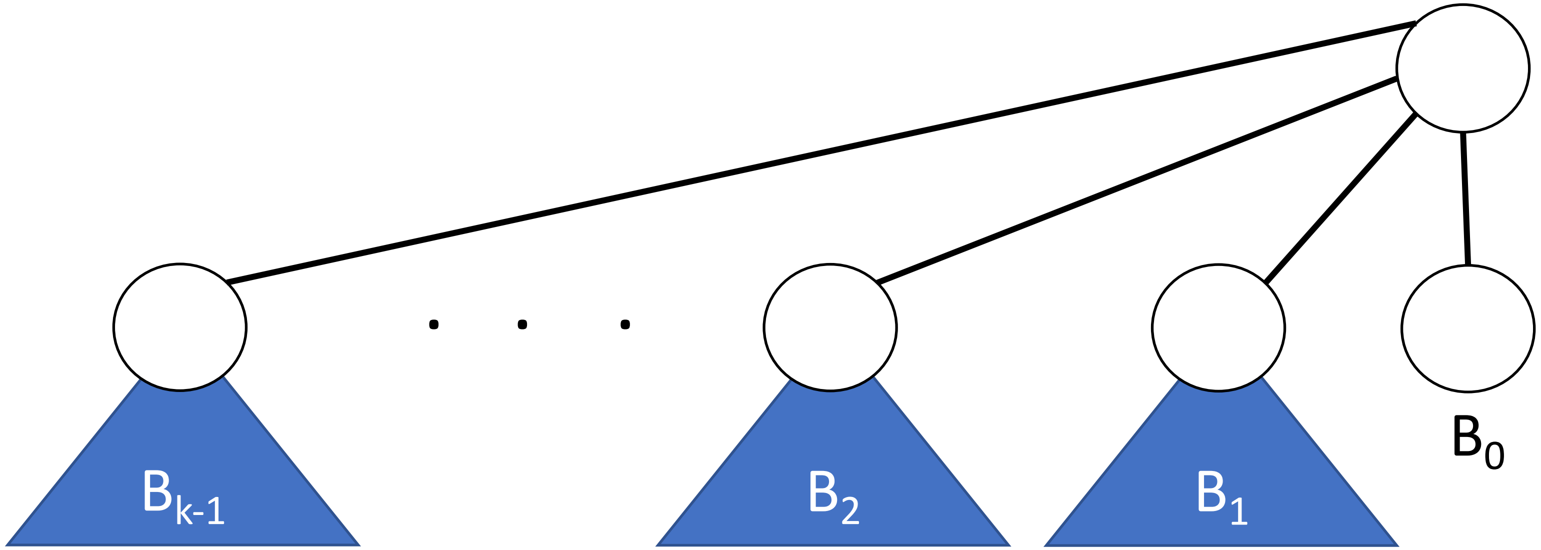
k = 0 B_0 :



k > = 1 B_k :



Binomial Tree B_k



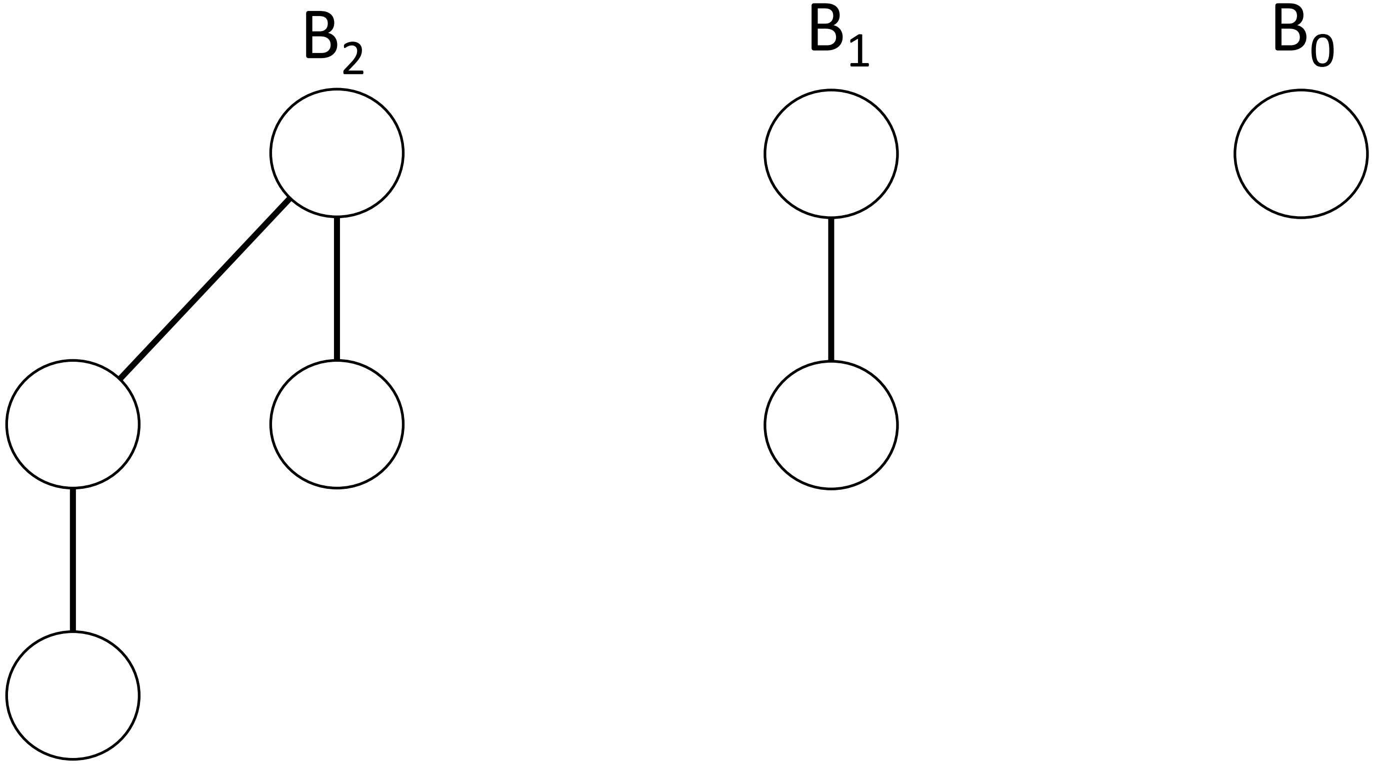
Binomial Forest F_n of size n

Sequence of B_k trees with *strictly decreasing* k's and a total of n nodes.

Example: Binomial Forest F_7 of n = 7 nodes

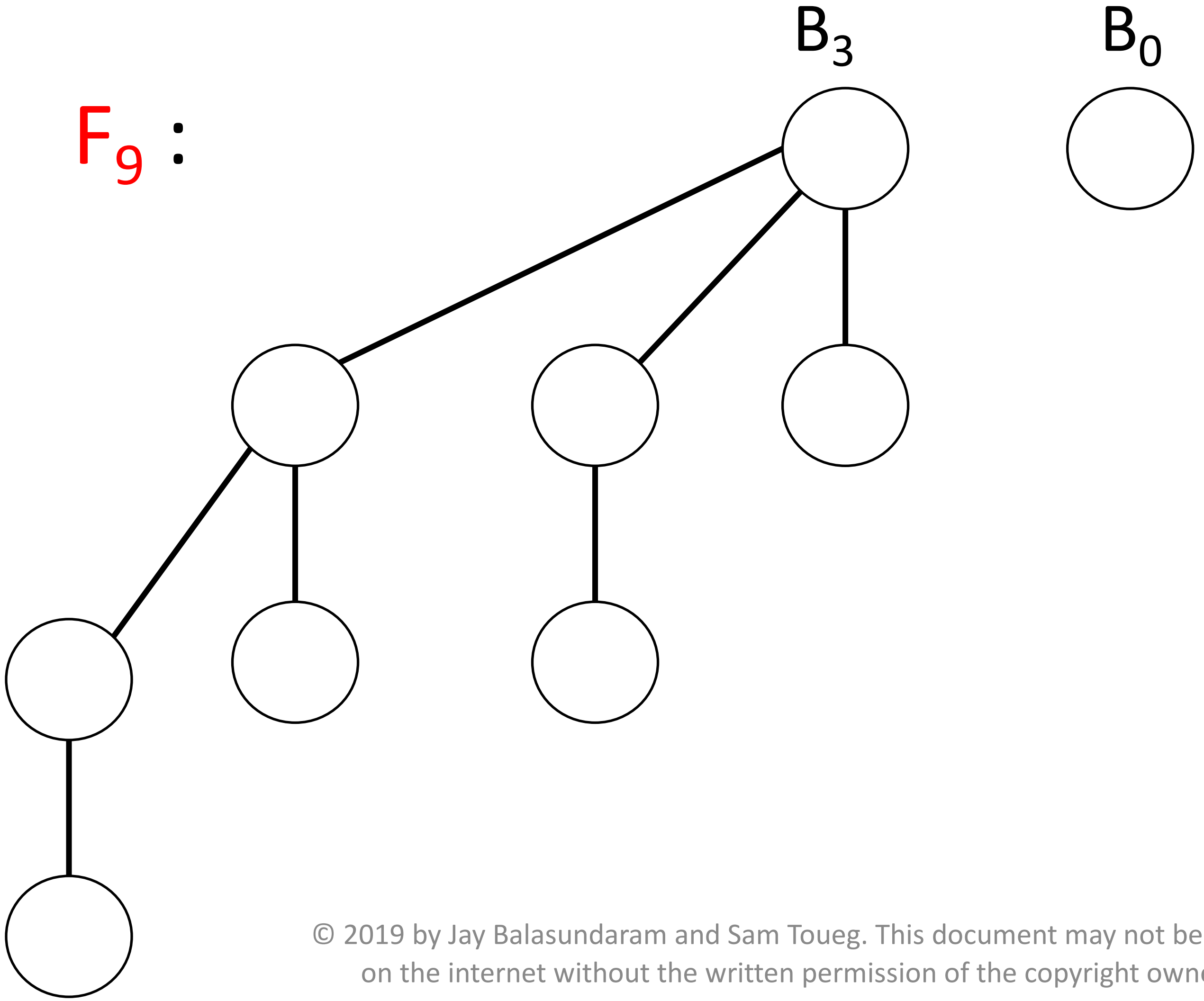
$n = 7 = \langle 1\ 1\ 1 \rangle_2 = 2^2 + 2^1 + 2^0$

F_7 :



Example: Binomial Forest F_9 of n = 9 nodes

$n = 9 = \langle 1\ 0\ 0\ 1 \rangle_2 = 2^3 + 2^0$

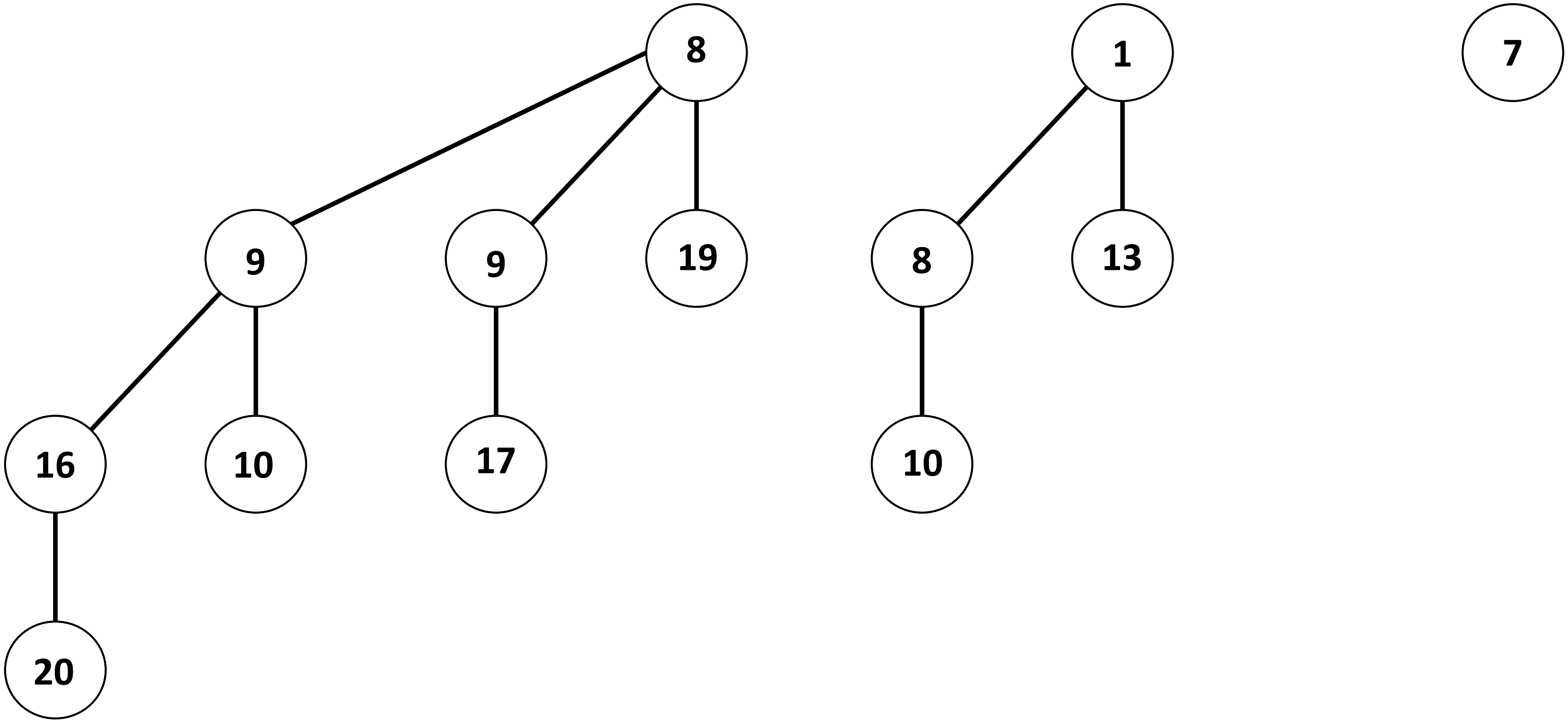


A Min Binomial Heap of n elements is a Binomial Forest F_n such that

1. Each node of F_n stores one element

2. Each B_k tree of F_n is Min-Heap ordered

Min Binomial Heap of size $n = 13 = \langle 1101 \rangle_2$



Binomial Heap Operations

Must implement the following operations:

- **Union(T, Q)**
- **Insert(T, x)**
- **Min(T)**
- **Extract_Min(T)**

High Level Ideas

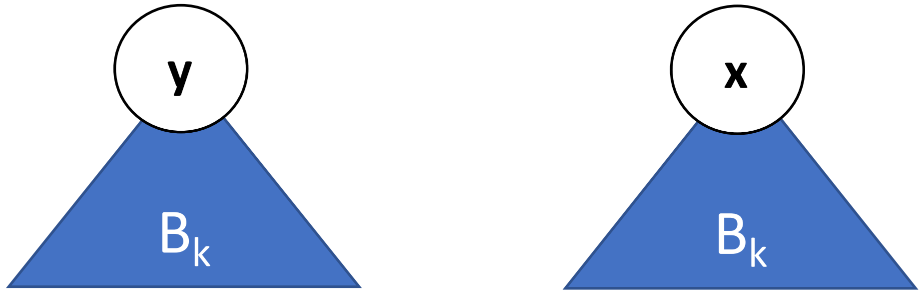
High Level Ideas

Lemma 1: Can merge two min heap-ordered B_k trees into a single min heap-ordered B_{k+1} tree with just one key-comparison.

High Level Ideas

Lemma 1: Can merge two min heap-ordered B_k trees into a single min heap-ordered B_{k+1} tree with just one key-comparison.

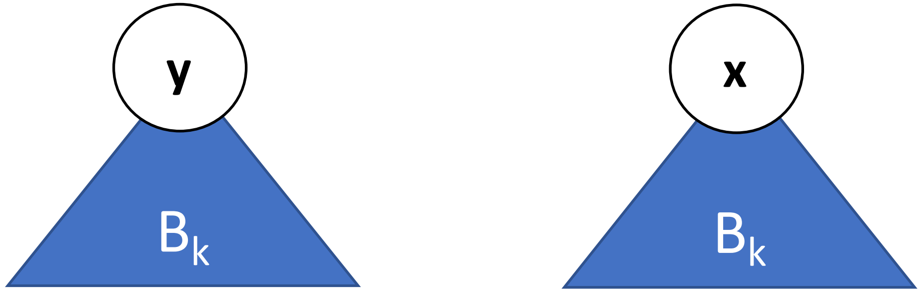
Proof: To merge



High Level Ideas

Lemma 1: Can merge two min heap-ordered B_k trees into a single min heap-ordered B_{k+1} tree with just one key-comparison.

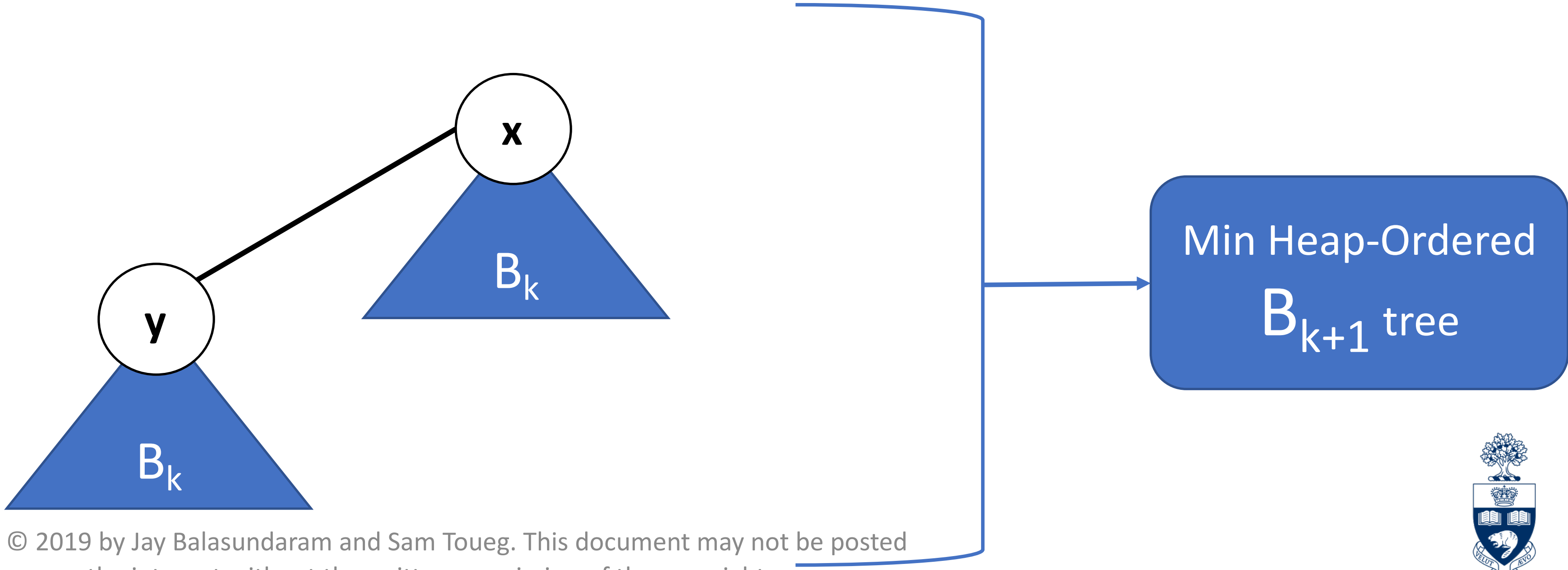
Proof: To merge, if x <= y



High Level Ideas

Lemma 1: Can merge two min heap-ordered B_k trees into a single min heap-ordered B_{k+1} tree with just one key-comparison.

Proof: To merge, if x <= y



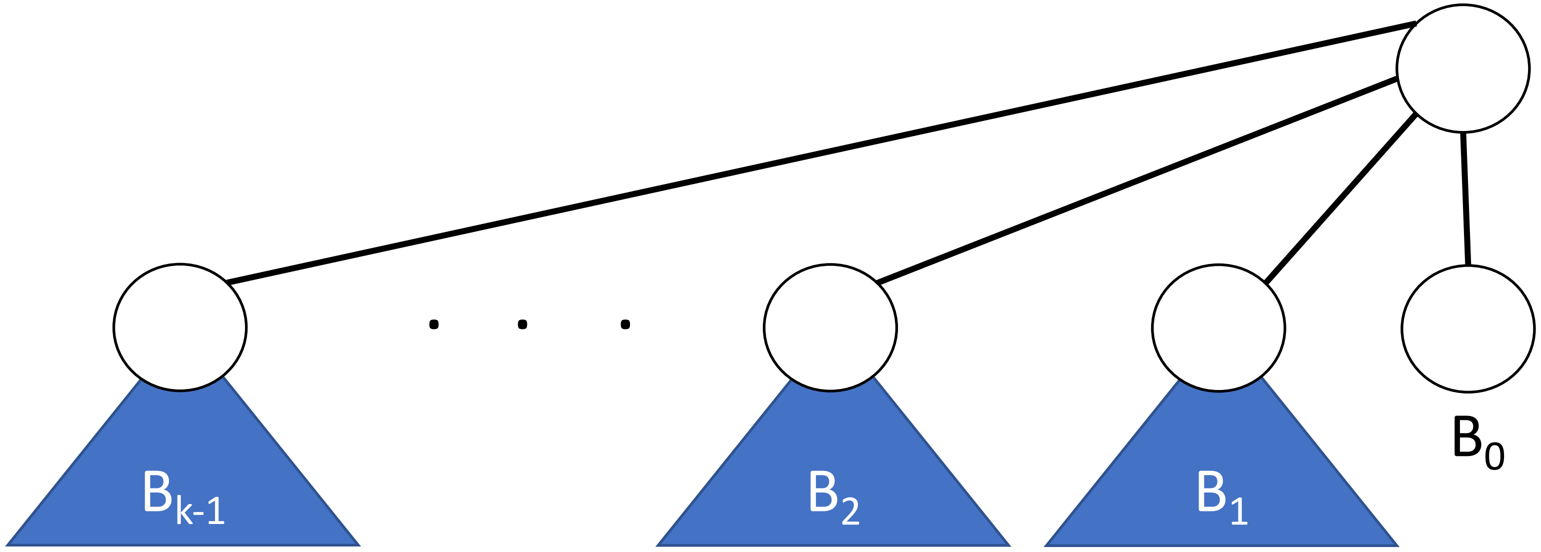
High Level Ideas

Lemma 2: Deleting the root of a min heap-ordered B_k tree gives a min binomial heap.

High Level Ideas

Lemma 2: Deleting the root of a min heap-ordered B_k tree gives a min binomial heap.

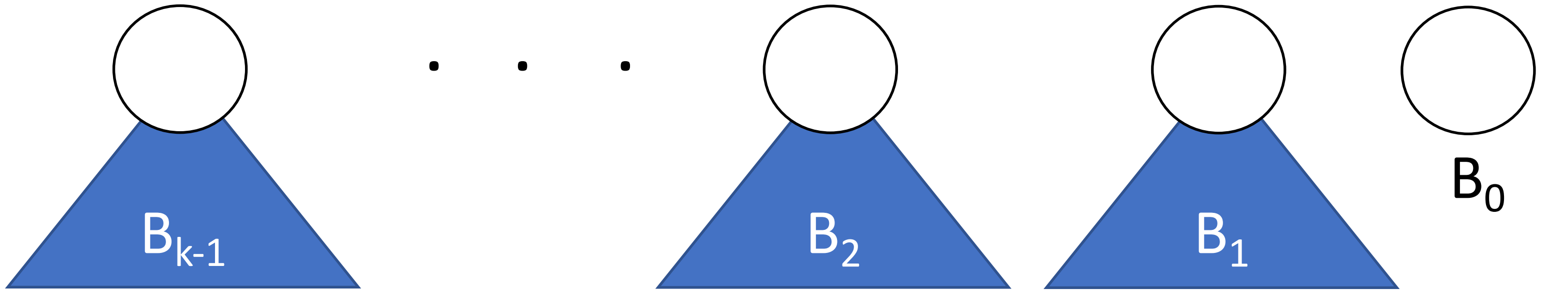
Proof: Deleting the root of min heap-ordered B_k tree.



High Level Ideas

Lemma 2: Deleting the root of a min heap-ordered B_k tree gives a min binomial heap.

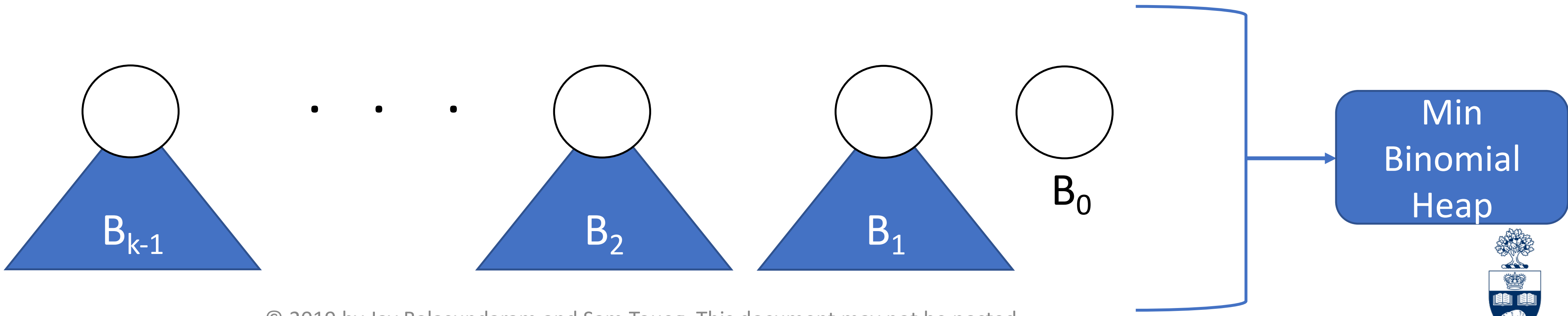
Proof: Deleting the root of min heap-ordered B_k tree.



High Level Ideas

Lemma 2: Deleting the root of a min heap-ordered B_k tree gives a min binomial heap.

Proof: Deleting the root of min heap-ordered B_k tree.



$S \leftarrow \text{Union}(T, Q)$

S ← Union(T, Q)

T is a Binomial Heap of size n = 3 = < 1 1$>_2$

1	1	B_1	B_0	:	T

$S \leftarrow$ Union(T, Q)

T is a Binomial Heap of size $n = 3 = <\ \ \ 1\ 1>_2$

Q is a Binomial Heap of size $n = 7 = <1\ 1\ 1>_2$

	1	1			B_1	B_0	: T
1	1	1		B_2	B_1	B_0	: Q

S ← Union(T, Q)

T is a Binomial Heap of size n = 3 = $<\ \ 1\ 1>_2$

Q is a Binomial Heap of size n = 7 = $<1\ 1\ 1>_2$

	1	1	+		B_1	B_0	:	T
1	1	1		B_2	B_1	B_0	:	Q

+

S ← Union(T, Q)

T is a Binomial Heap of size n = 3 = < 1 1>$_2$

Q is a Binomial Heap of size n = 7 = < 1 1 1>$_2$

	1				B_1			**Carry**
	1	1	**+**		B_1	B_0	: T	**+**
1	1	1		B_2	B_1	B_0	: Q	
		0				X	: S	

S ← Union(T, Q)

T is a Binomial Heap of size n = 3 = <   1 1>$_2$

Q is a Binomial Heap of size n = 7 = < 1 1 1>$_2$

1	1				B_2	B_1			**Carry**
	1	1	**+**			B_1	B_0	:	T
									+
1	1	1			B_2	B_1	B_0	:	Q
	1	0				B_1	X	:	S

S ← Union(T, Q)

T is a Binomial Heap of size n = 3 = < 1 1$>_2$

Q is a Binomial Heap of size n = 7 = < 1 1 1$>_2$

1	1	1			B_3	B_2	B_1			**Carry**
		1	1				B_1	B_0	:	T
				+						+
	1	1	1			B_2	B_1	B_0	:	Q
1	0	1	0		B_3	X	B_1	X	:	S

S ← Union(T, Q)

T is a Binomial Heap of size $n = 3 = <\ \ 1\ 1>_2$

Q is a Binomial Heap of size $n = 7 = < 1\ 1\ 1>_2$

1	1	1			B_3	B_2	B_1		**Carry**
		1	1	**+**			B_1	B_0	: T
									+
	1	1	1			B_2	B_1	B_0	: Q
1	0	1	0		B_3	X	B_1	X	: S

How many new edges were added?

$S \leftarrow$ Union(T, Q)

T is a Binomial Heap of size $n = 3 = <\ \ 1\ 1>_2$

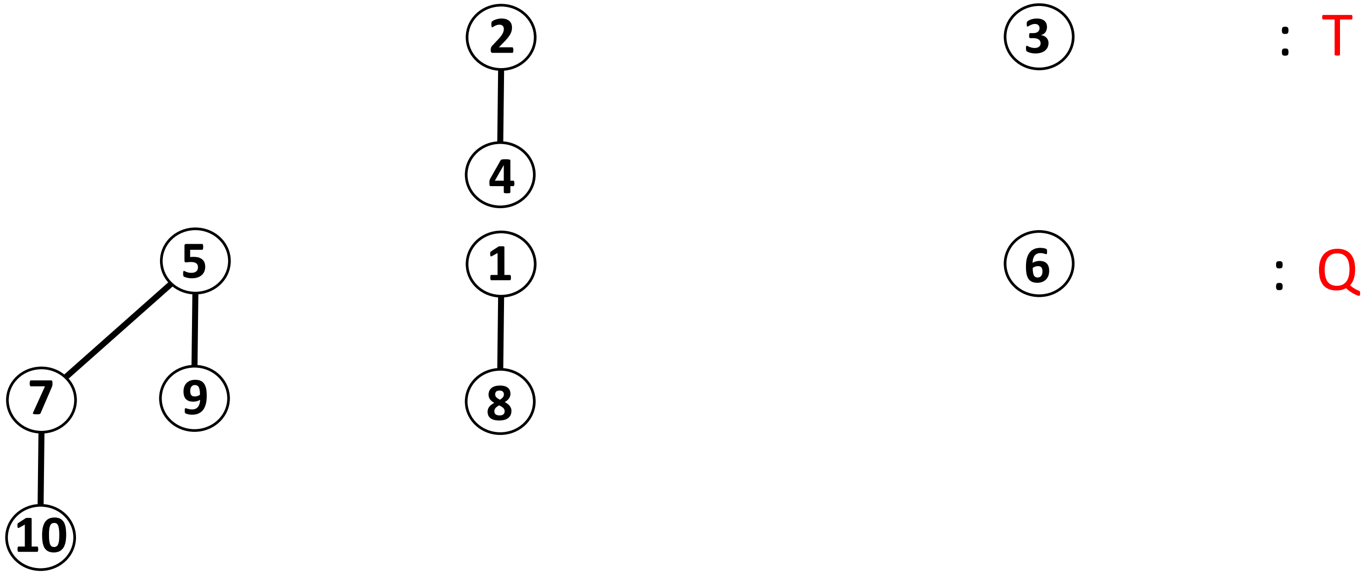
Q is a Binomial Heap of size $n = 7 = <1\ 1\ 1>_2$

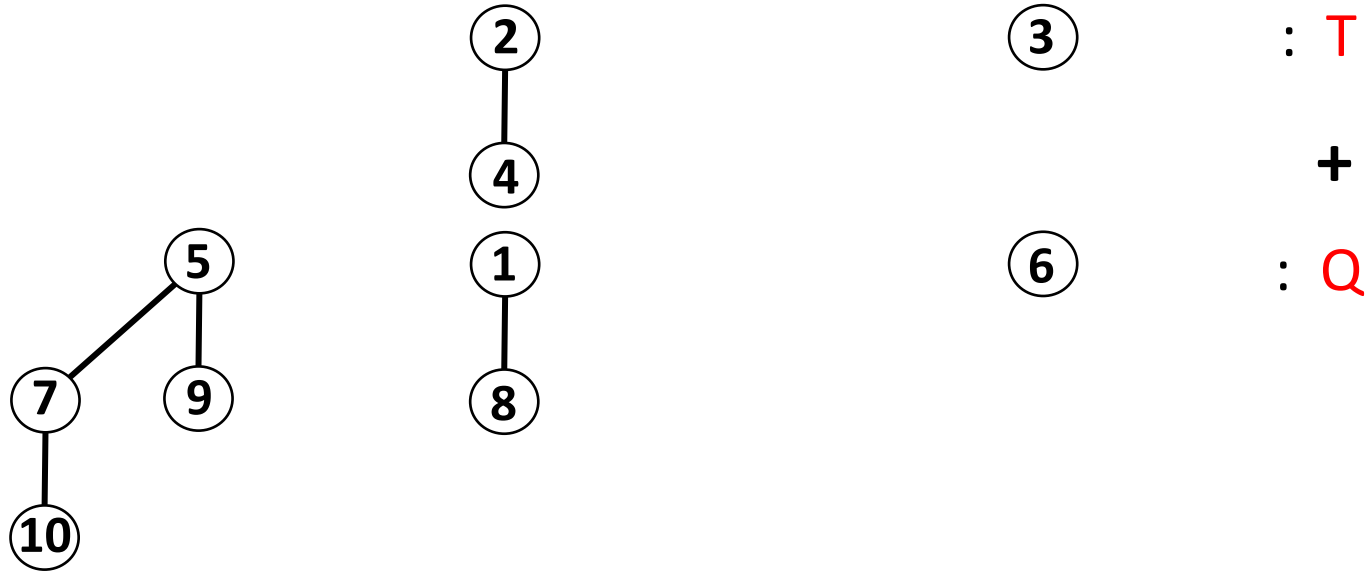
1	1	1		B_3	B_2	B_1		Carry	
		1	1			B_1	B_0	:	T
			+						+
	1	1	1		B_2	B_1	B_0	:	Q
1	0	1	0	B_3	X	B_1	X	:	S

How many new edges were added?

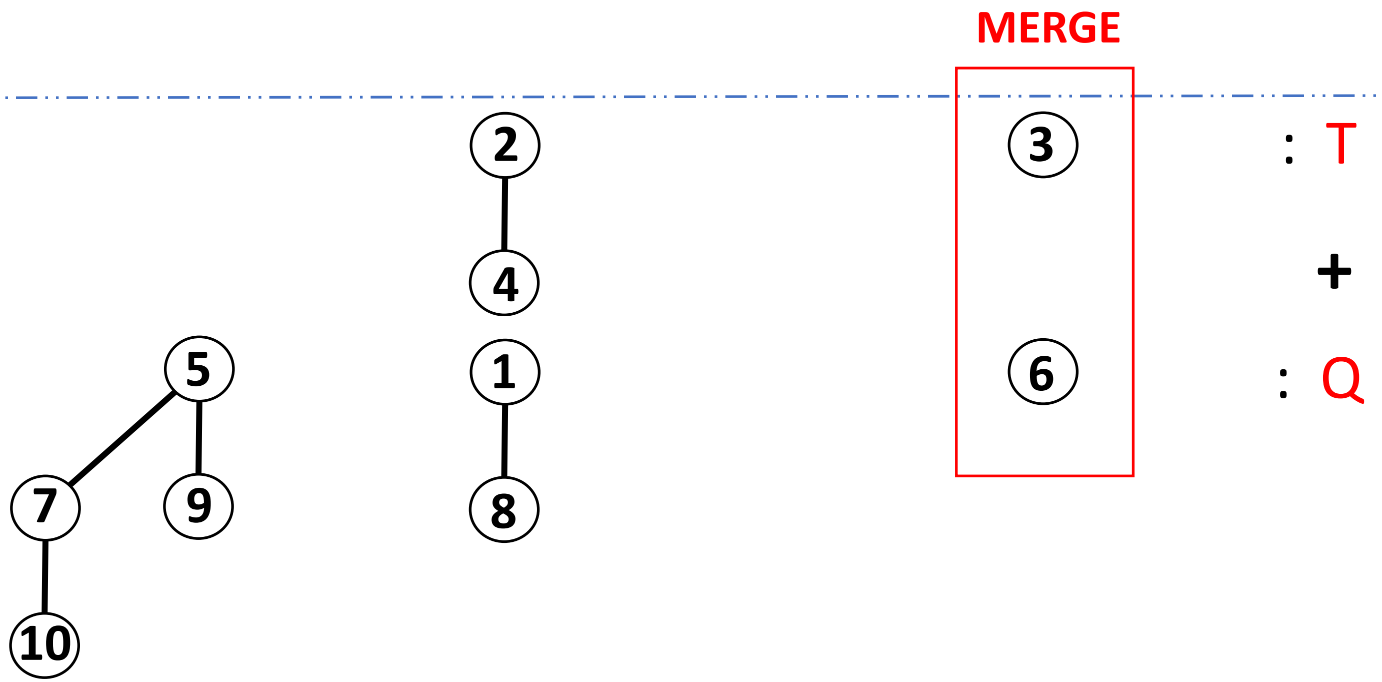
3

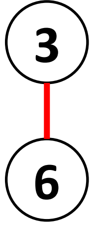

2
3
: T
4


2
4
3
: T
5
7
9
10
1
8
6
: Q

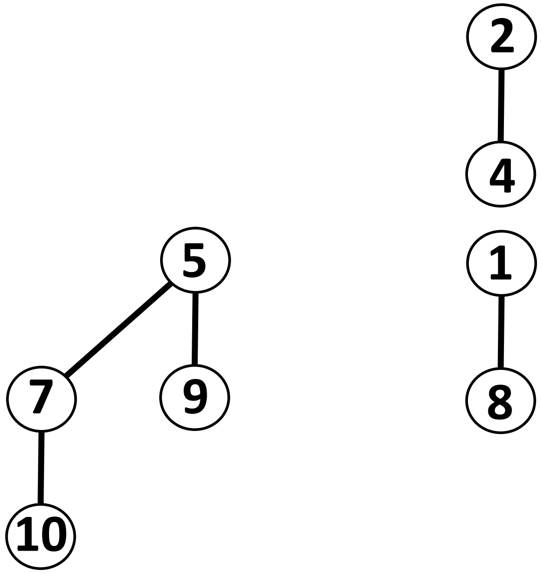


: T
+
: Q


MERGE
2
4
1
8
5
7
9
10
3
6
: T
+
: Q



Carry



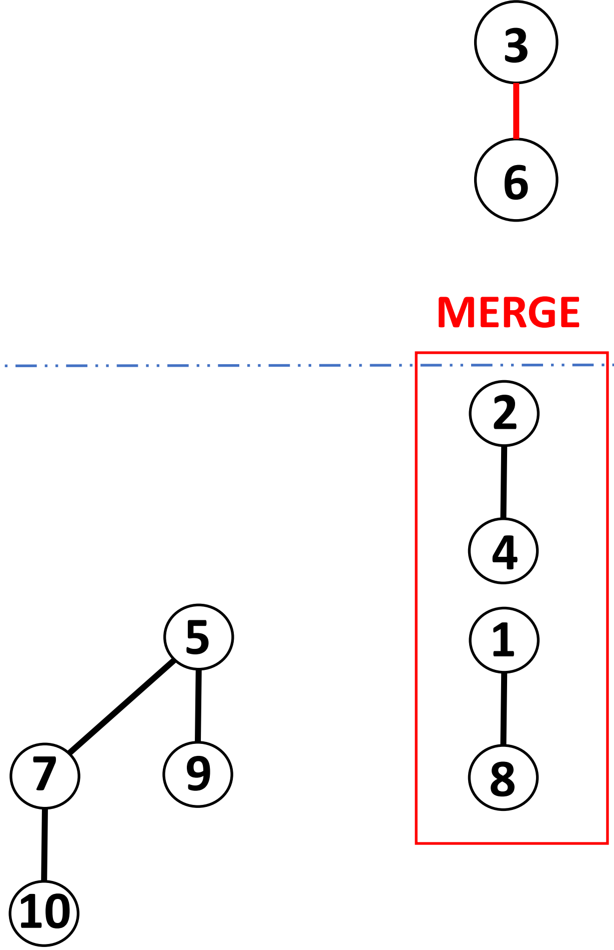
3 : T

+

6 : Q

X : S

Carry

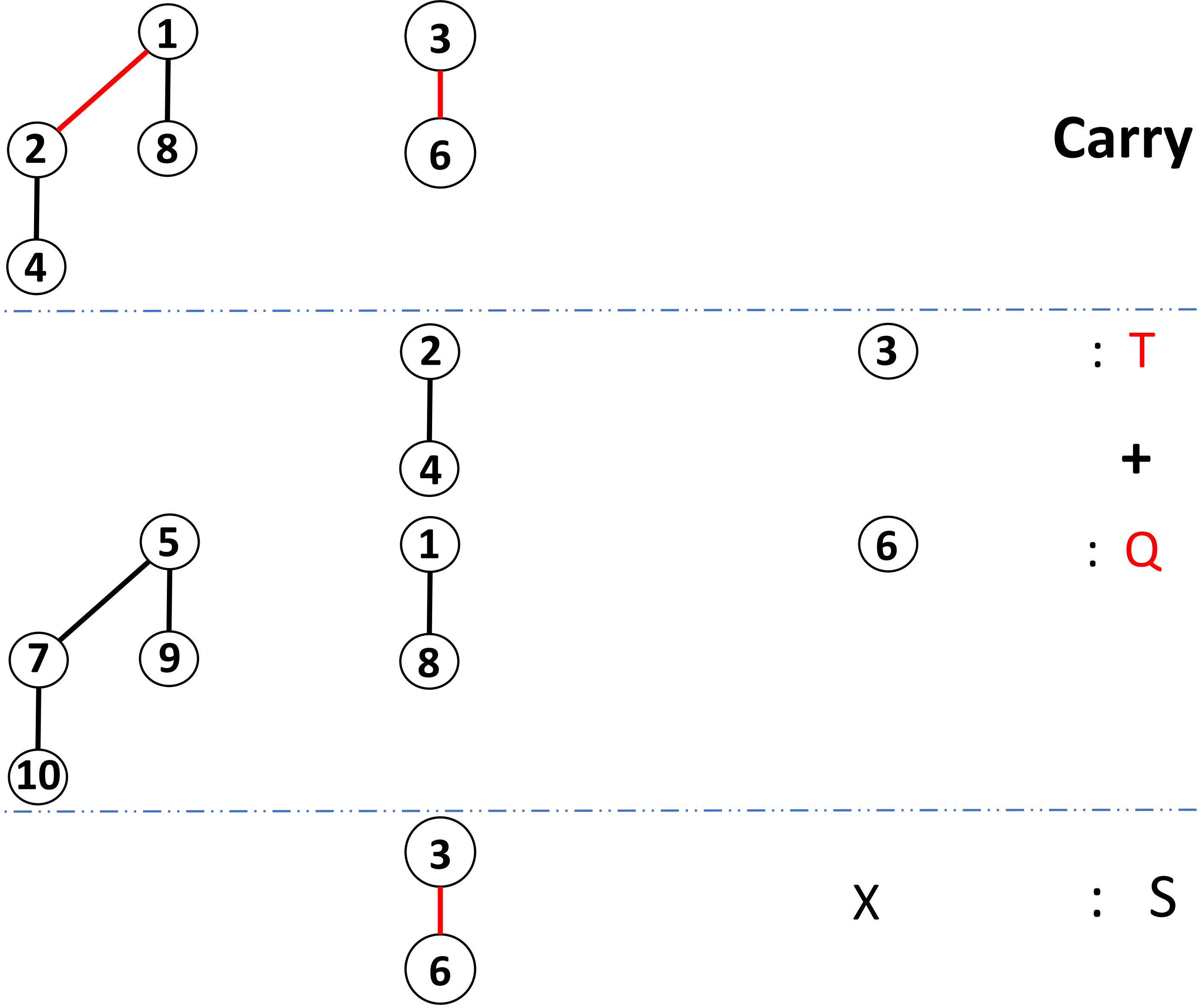


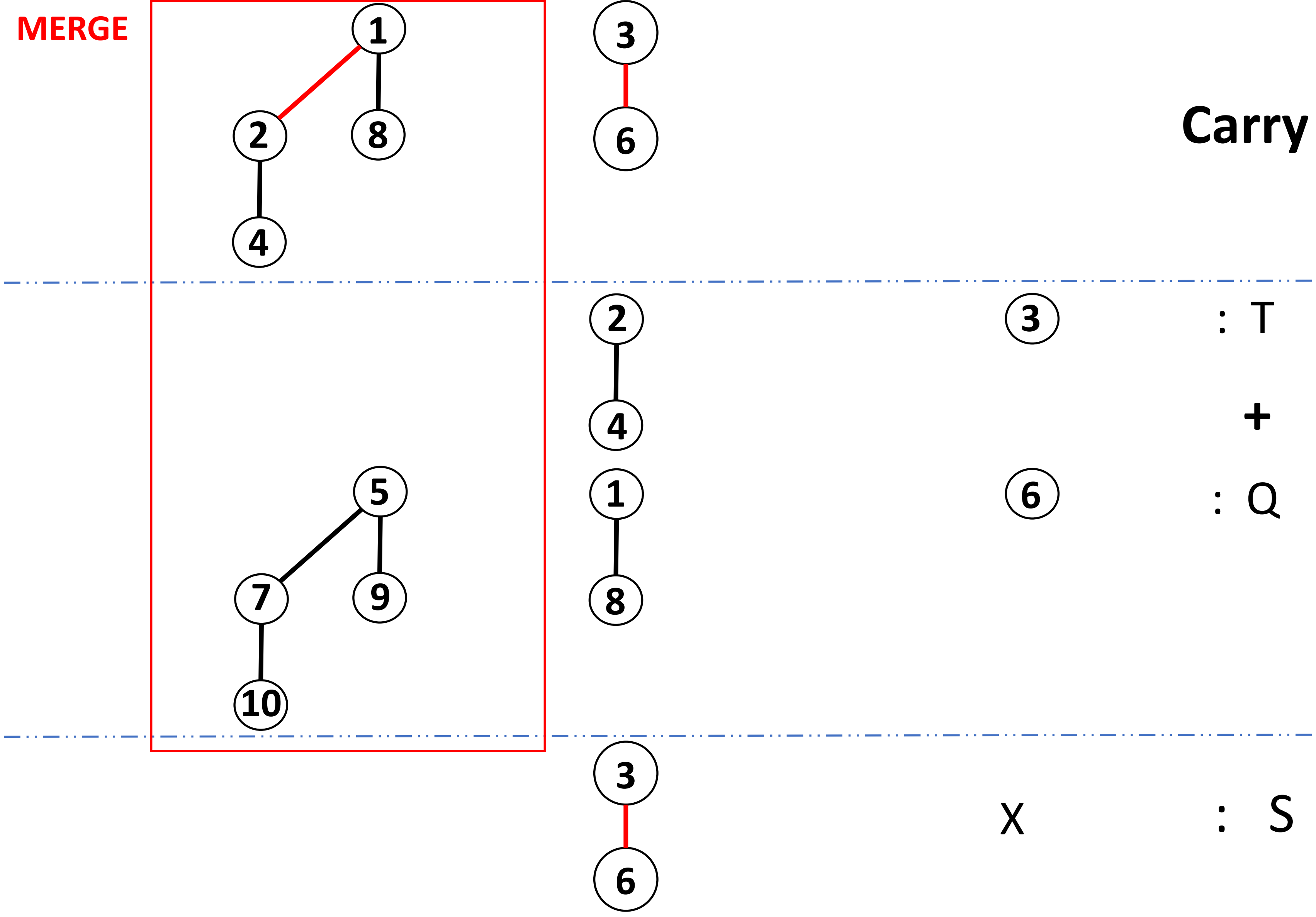
3 : T

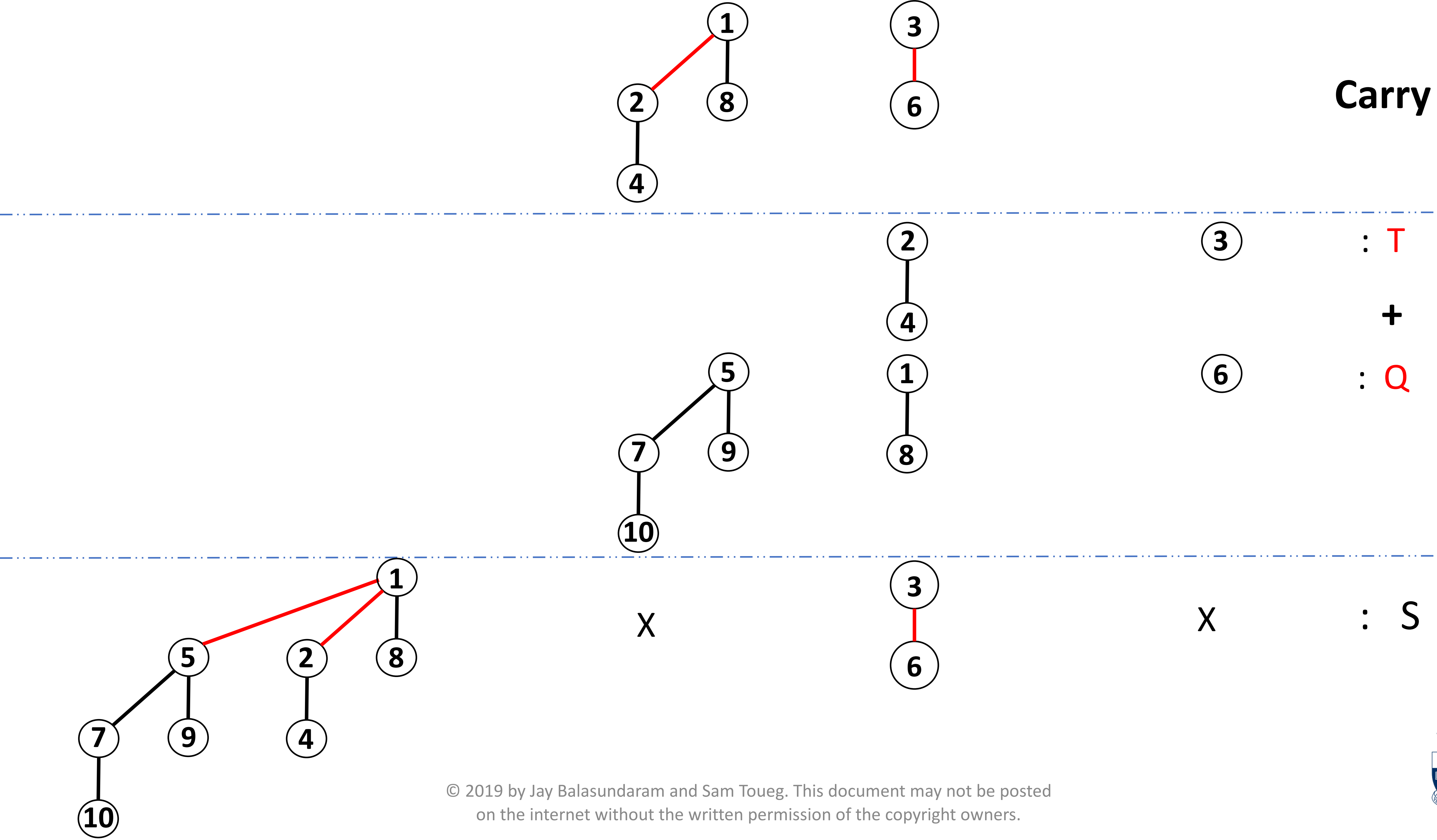
+

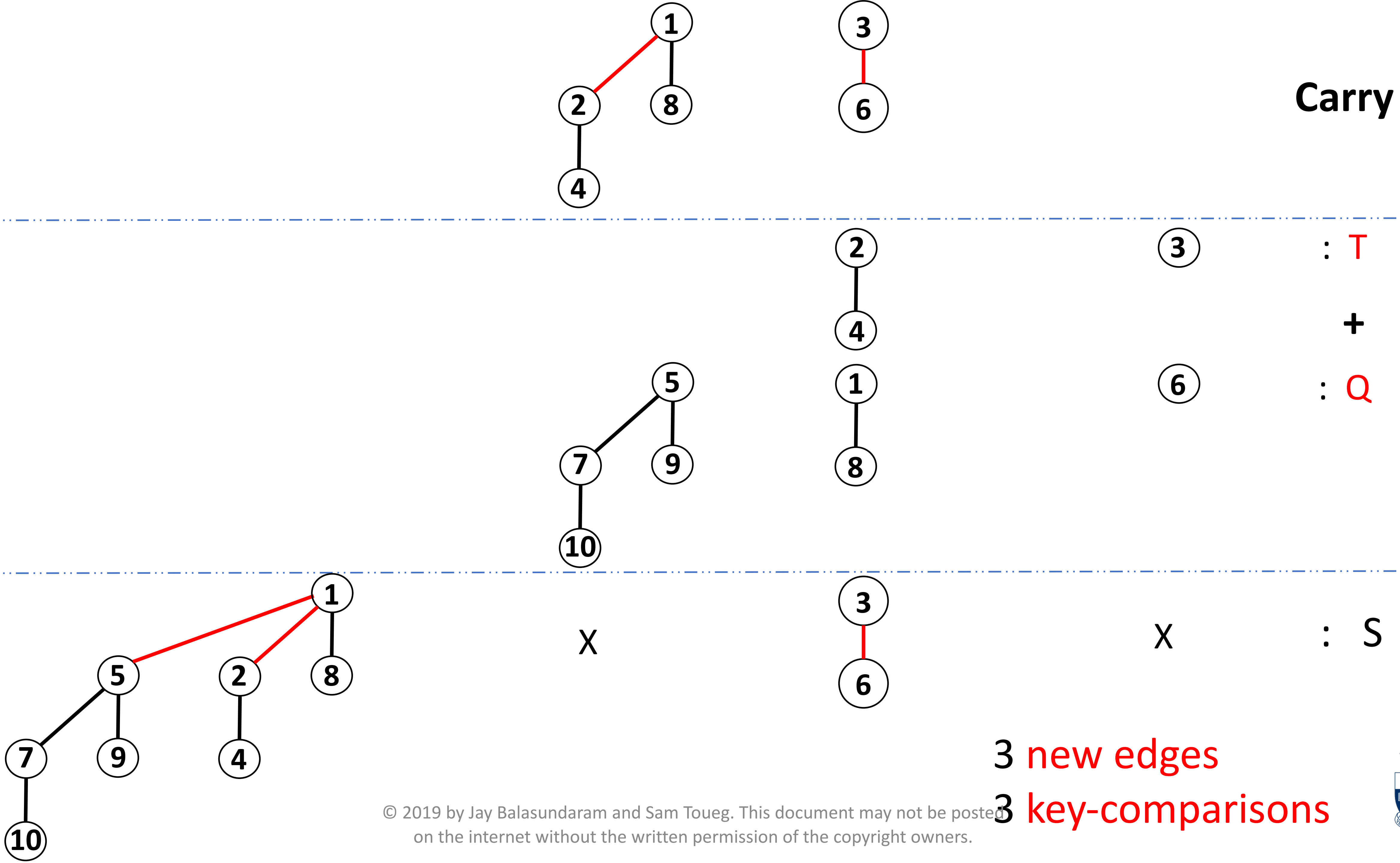
6 : Q

X : S


1
2
8
4
3
6
Carry
2
4
3
: T
+
5
1
6
: Q
7
9
8
10
3
6
X
: S


MERGE
Carry
1
2
8
4
3
6
5
7
9
10
2
4
1
8
3
6
3 : T
+
6 : Q
X : S


Carry
: T
+
: Q
X
X
: S



3 new edges

3 key-comparisons

Worst-Case Complexity of **Union(T, Q)**

Say |T| <= n and |Q| <= n (i.e. each contains at most n elements)

Worst-Case Complexity of **Union(T, Q)**

Say |T| <= n and |Q| <= n (i.e. each contains at most n elements)

$\Rightarrow$ Each of T, Q have O(log n) B_k trees.

Worst-Case Complexity of **Union(T, Q)**

Say |T| <= n and |Q| <= n (i.e. each contains at most n elements)

$\Rightarrow$ Each of T, Q have O(log n) B_k trees.

$\Rightarrow$ **Union**(T, Q) takes at most O(log n) key-comparisons

$S \leftarrow \text{Insert}(T, x)$

S ← Insert(T, x)

S ←**Union**(T, {x})

S ← Insert(T, x)

S ←**Union**(T, {x})

Trivial Binomial Heap containing just x

$S \leftarrow$ Insert(T, x)

$S \leftarrow$ **Union**(T, {x})

Trivial Binomial Heap containing just x

If |T| <= n, **Insert**(T, x) takes at most O(log n) key-comparisons

Min(T)

Min(T)

Scan the roots of the B_k trees of T and return the smallest key.

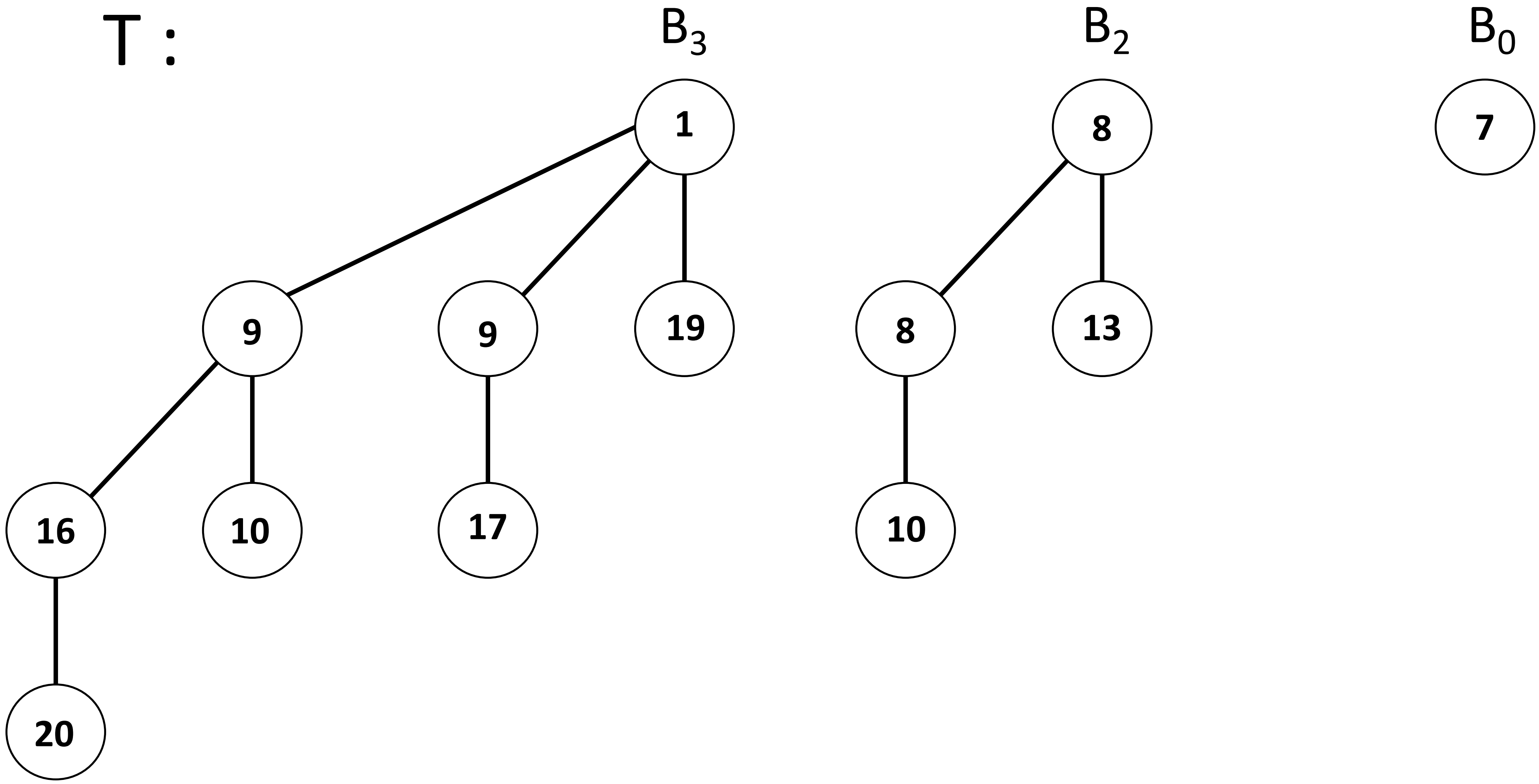
Min(T)

Scan the roots of the B_k trees of T and return the smallest key.

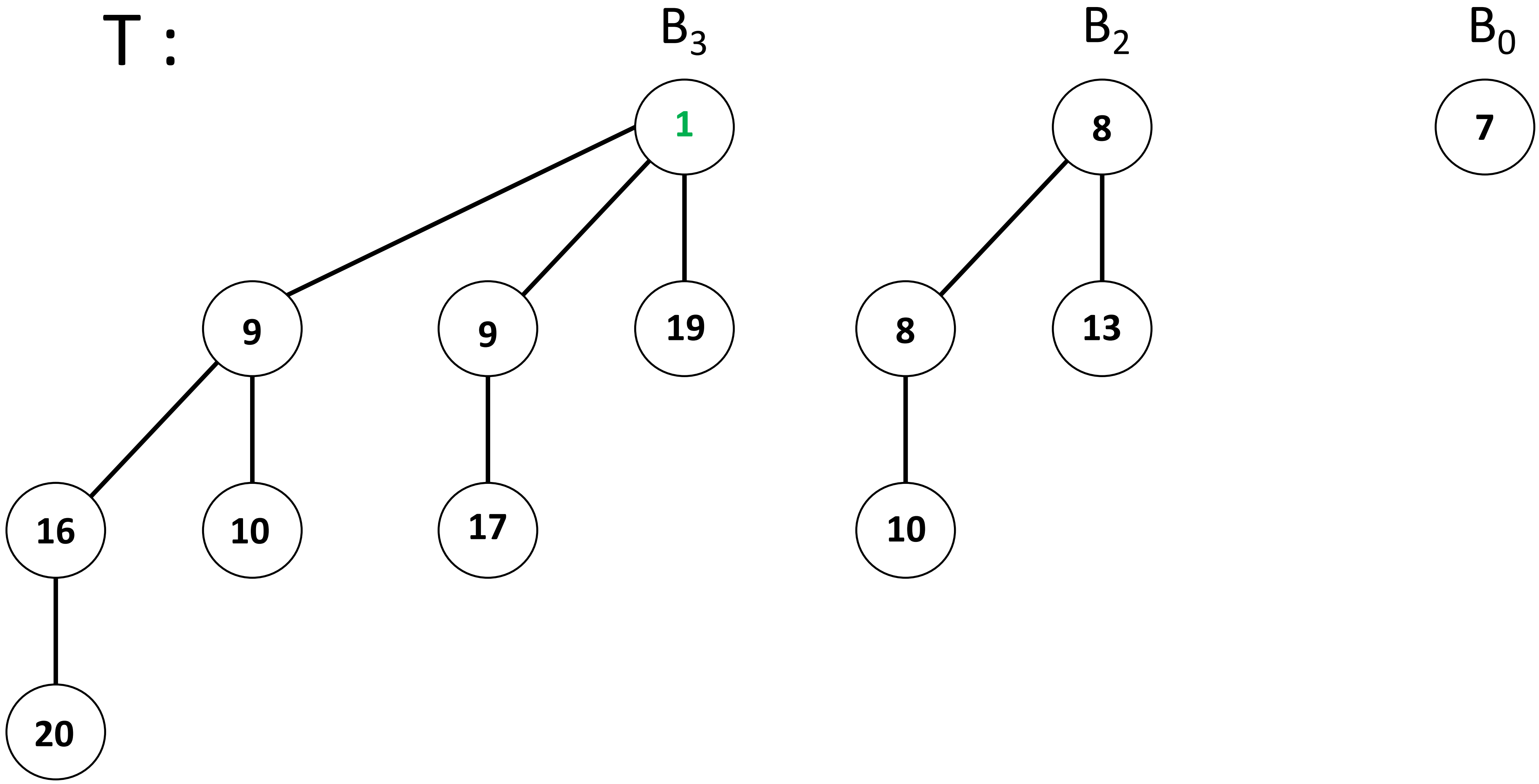
If |T| <= n, Min(T) takes at most O(log n) key-comparisons

Extract_Min(T)

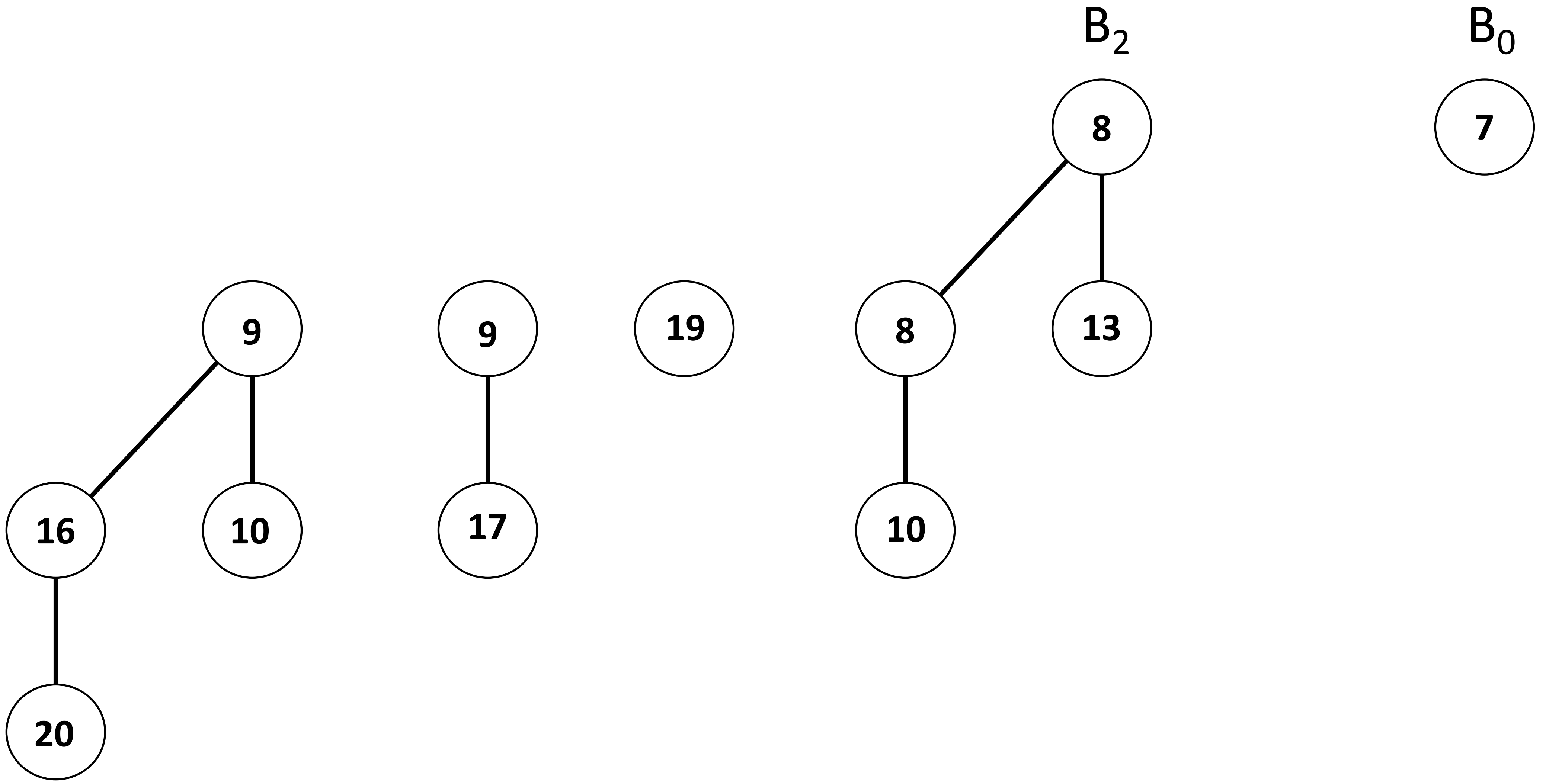
Extract_Min(T)



Extract_Min(T)

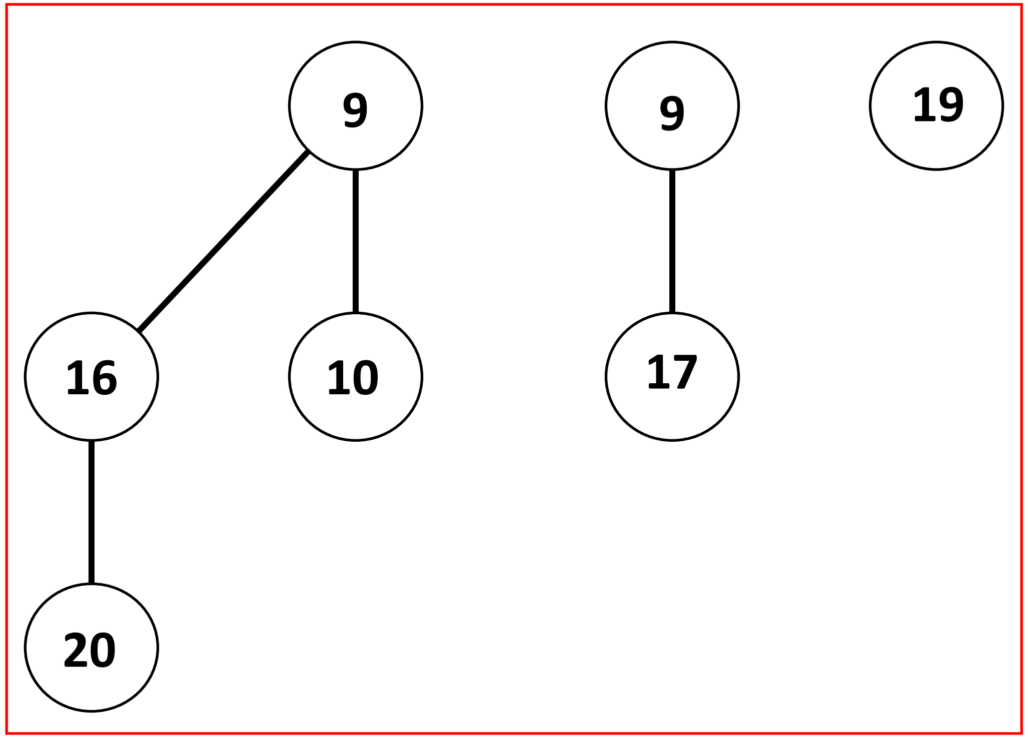


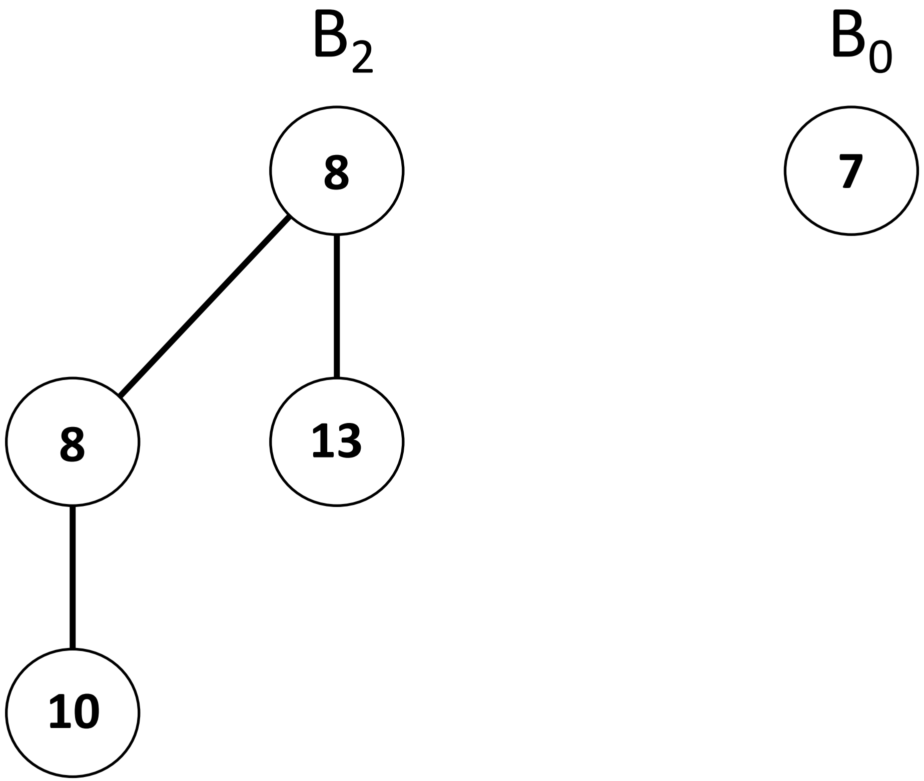
Extract_Min(T)



Extract_Min(T)

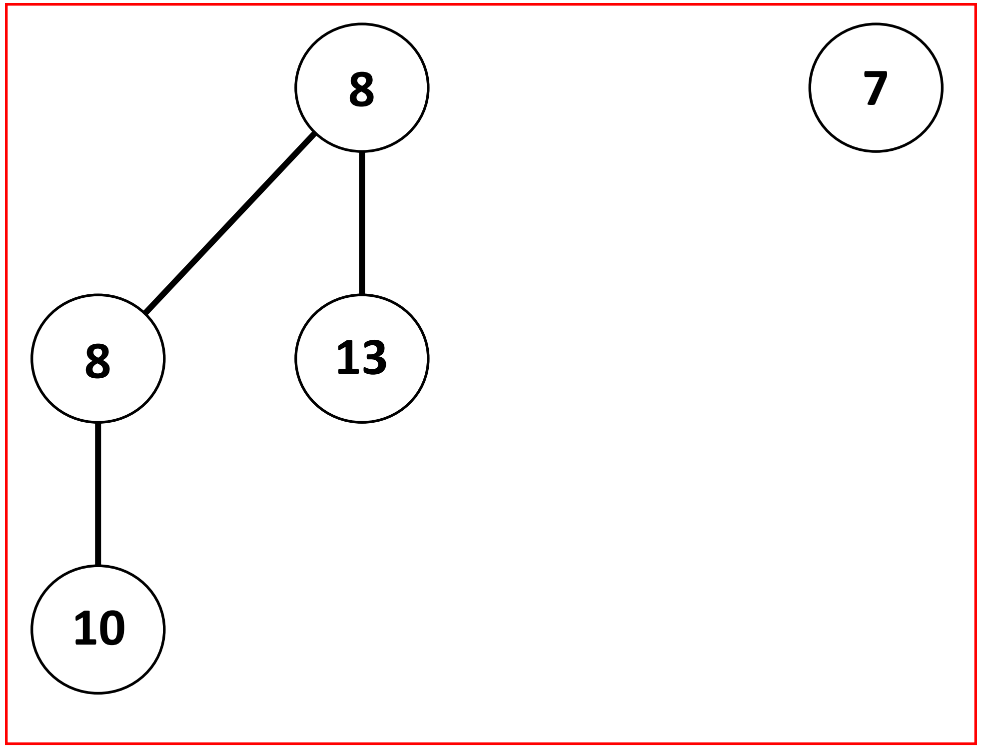
Binomial Heap S = $B_3 - \{1\}$



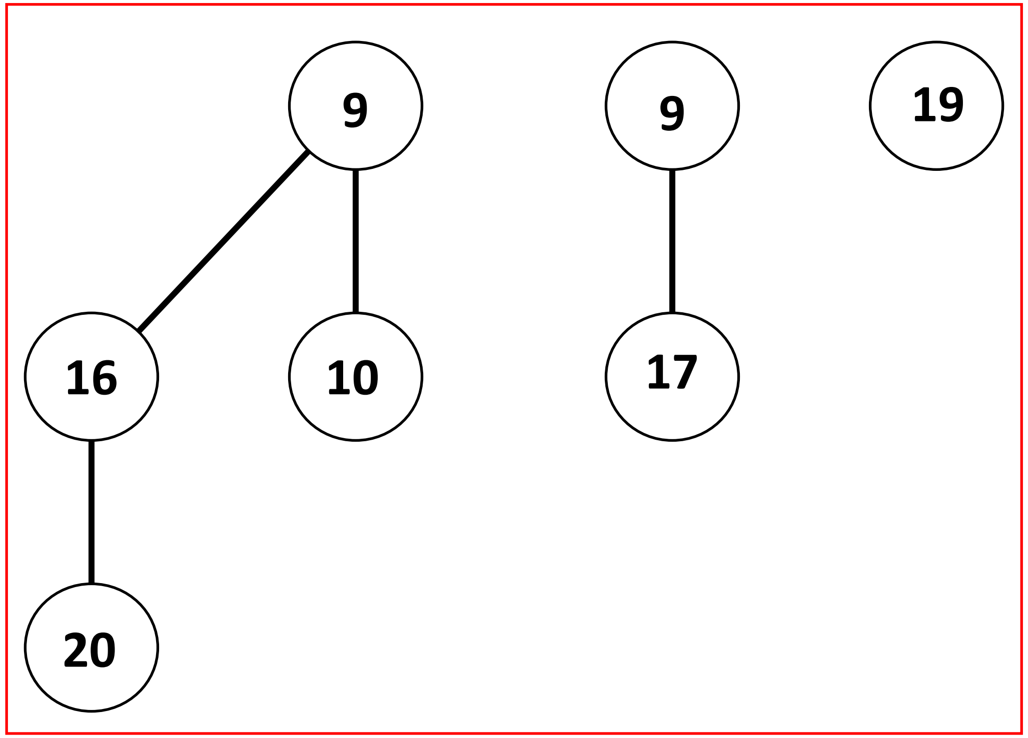


Extract_Min(T)

Binomial Heap U = T - B_3

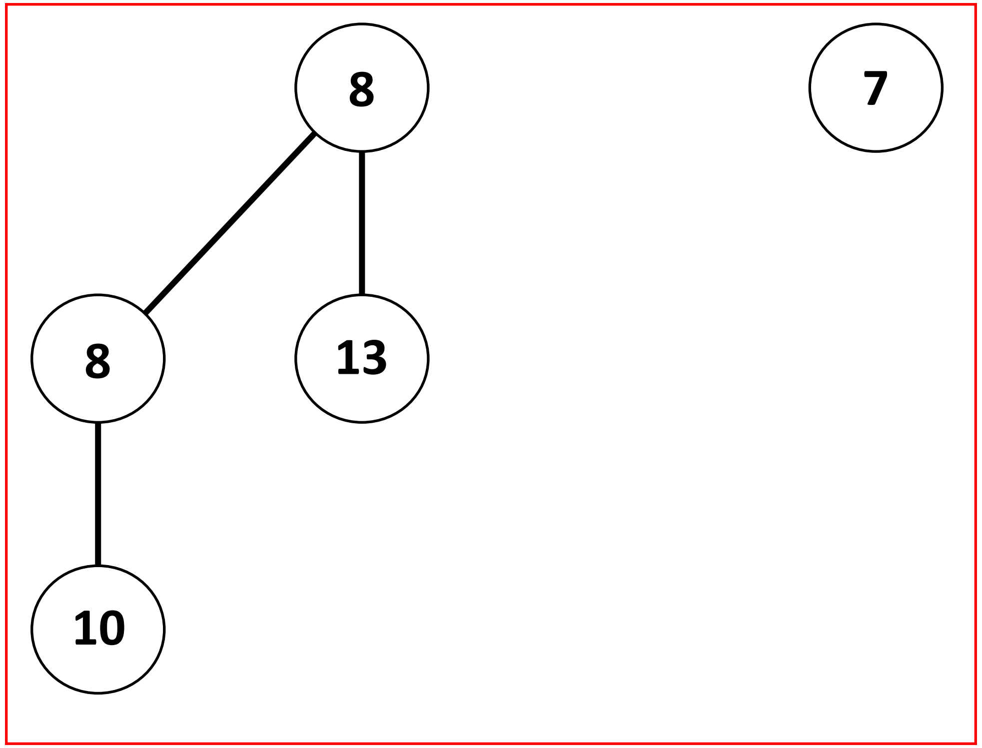


Binomial Heap S = B_3 – {1}

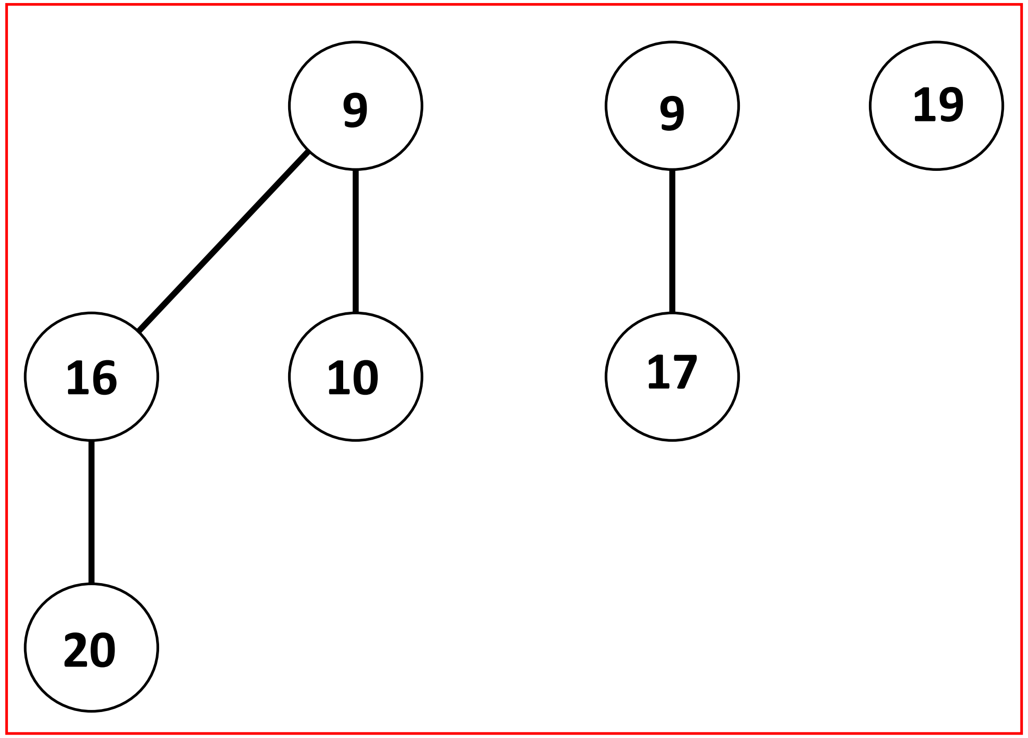


Extract_Min(T)

Binomial Heap U = T - B_3



Binomial Heap S = $B_3 - \{1\}$



Now do: T ← **Union**(U,S)

Extract_Min(T)

- Do **Min**(T) to locate the smallest element – Say it is the root of B_i

$$U = T - B_i$$

Extract_Min(T)

- Do **Min**(T) to locate the smallest element – Say it is the root of B_i

$$U = T - B_i$$

- Delete root of B_i. By Lemma 2, we get a Binomial Heap S, where

$$S = B_i – (\text{root of } B_i)$$

Extract_Min(T)

- Do **Min**(T) to locate the smallest element – Say it is the root of B_i

  $U = T - B_i$

- Delete root of B_i. By Lemma 2, we get a Binomial Heap S, where

  $S = B_i$ – (root of B_i)

- T ← **Union(**U, S**)**

Extract_Min(T)

- Do **Min**(T) to locate the smallest element – Say it is the root of B_i

  $U = T - B_i$

- Delete root of B_i. By Lemma 2, we get a Binomial Heap S, where

  $S = B_i$ – (root of B_i)

- T ← **Union(**U, S**)**

If |T| <= n, **Extract_Min(T)** takes at most O(log n) key-comparisons

A couple more operations

- Given pointer to a node x in a Binomial Heap T, you can do:

A couple more operations

- Given pointer to a node x in a Binomial Heap T, you can do:

Decrease_Key(T, x, k): Decrease the key at node x to k.

A couple more operations

- Given pointer to a node x in a Binomial Heap T, you can do:

  Decrease_Key(T, x, k): Decrease the key at node x to k.

  Remove(T, x): Remove the key at node x.

A couple more operations

- Given pointer to a node x in a Binomial Heap T, you can do:

Decrease_Key(T, x, k): Decrease the key at node x to k.

Remove(T, x): Remove the key at node x.

Both in O(log n) time

A couple more operations

- Given pointer to a node x in a Binomial Heap T, you can do:

  Decrease_Key(T, x, k): Decrease the key at node x to k.
  Remove(T, x): Remove the key at node x.

  Both in O(log n) time

- How do you do **Increase_Key**(T, x, k) ?

Cost of k successive inserts

- T: Binomial Heap with n elements.

- Cost of k successive inserts into T ?

Cost of k successive inserts

- T: Binomial Heap with n elements.

- Cost of k successive inserts into T ?

Insert(T, x_1) , **Insert**(T, x_2) , ... , **Insert**(T, x_k)

Cost of k successive inserts

- T: Binomial Heap with n elements.
- Cost of k successive inserts into T ?

Insert(T, x_1) , **Insert**(T, x_2) , ... , **Insert**(T, x_k)

O(log n)

Cost of k successive inserts

- T: Binomial Heap with n elements.

- Cost of k successive inserts into T ?

Insert(T, x_1) , **Insert**(T, x_2) , ... , **Insert**(T, x_k)

O(log n)   O(log (n + 1))

Cost of k successive inserts

- T: Binomial Heap with n elements.

- Cost of k successive inserts into T ?

Insert(T, x_1) , **Insert**(T, x_2) , ... , **Insert**(T, x_k)

$O(\log n)$ $O(\log (n + 1))$ . . . . . $O(\log (n + k))$

Cost of k successive inserts

- T: Binomial Heap with n elements.

- Cost of k successive inserts into T ?

Insert(T, x_1) , **Insert**(T, x_2) , ... , **Insert**(T, x_k)

O(log n)   O(log (n + 1)) . . . . .   O(log (n + k))

- Total: O(k log (n + k))

Cost of k successive inserts

- T: Binomial Heap with n elements.

- Cost of k successive inserts into T ?

Insert(T, x_1) , **Insert**(T, x_2) , ... , **Insert**(T, x_k)

O(log n)   O(log (n + 1)) . . . . .   O(log (n + k))

- Total: O(k log (n + k))

- Is the cost of k successive inserts actually lower?

Cost of k successive inserts

- T: Binomial Heap with n elements.

- Cost of k successive inserts into T ?

Insert(T, x_1) , **Insert**(T, x_2) , ... , **Insert**(T, x_k)

O(log n)   O(log (n + 1)) . . . . .   O(log (n + k))

- Total: O(k log (n + k))

- Is the cost of k successive inserts actually lower? **Yes!**

Example: Say $|T| = 27 = < 11011 >_2$ $\quad$ $T = < B_4\ B_3\ B_1\ B_0 >$

Example: Say $|T| = 27 = \langle 11011 \rangle_2$ $\qquad T = \langle B_4\ B_3\ B_1\ B_0 \rangle$

$$
\begin{array}{lccccc}
T & 1 & 1 & 0 & 1 & 1 \\
+ & & & & & \\
x_1 & & & & & 1
\end{array}
$$

Example: Say $|T| = 27 = < 11011 >_2$ $\quad$ $T = < B_4\ B_3\ B_1\ B_0 >$

$$
\begin{array}{lccccc}
 & & & & \color{red}{1} & \\
\hline
T & 1 & 1 & 0 & 1 & 1 \\
+ & & & & & \\
x_1 & & & & & 1 \\
\hline
 & & & & & 0
\end{array}
$$

Example: Say $|T| = 27 = < 11011 >_2$ $\quad T = < B_4\ B_3\ B_1\ B_0 >$

$$
\begin{array}{lccccc}
 & & & 1 & 1 & \\
T & 1 & 1 & 0 & 1 & 1 \\
+ & & & & & \\
x_1 & & & & & 1 \\
\hline
 & & & & 0 & 0
\end{array}
$$

Example: Say $|T| = 27 = < 11011 >_2$ $T = < B_4\ B_3\ B_1\ B_0 >$

$$
\begin{array}{lccccc}
 & & & 1 & 1 & \\
\hline
T & 1 & 1 & 0 & 1 & 1 \\
+ & & & & & \\
x_1 & & & & & 1 \\
\hline
 & 1 & 1 & 1 & 0 & 0
\end{array}
$$

Example: Say $|T| = 27 = \langle 11011 \rangle_2$ $\quad$ $T = \langle B_4\ B_3\ B_1\ B_0 \rangle$

$$
\begin{array}{lrrrrr}
 & & & 1 & 1 & \\
\hline
T & 1 & 1 & 0 & 1 & 1 \\
+ & & & & & \\
x_1 & & & & & 1 \\
\hline
 & 1 & 1 & 1 & 0 & 0
\end{array}
$$

Cost : 2

Example: Say $|T| = 27 = \langle 11011 \rangle_2$ $T = \langle B_4\ B_3\ B_1\ B_0 \rangle$

$$
\begin{array}{lccccc}
 & & & 1 & 1 & \\
\hline
T & 1 & 1 & 0 & 1 & 1 \\
+ & & & & & \\
x_1 & & & & & 1 \\
\hline
 & 1 & 1 & 1 & 0 & 0
\end{array}
$$

Cost : 2

$$
\begin{array}{lccccc}
\hline
T & 1 & 1 & 1 & 0 & 0 \\
+ & & & & & \\
x_2 & & & & & 1 \\
\hline
\end{array}
$$

Example: Say $|T| = 27 = \langle 11011 \rangle_2$ $T = \langle B_4\ B_3\ B_1\ B_0 \rangle$

$$\begin{array}{lccccc} & & & 1 & 1 & \\ \hline T & 1 & 1 & 0 & 1 & 1 \\ + & & & & & \\ x_1 & & & & & 1 \\ \hline & 1 & 1 & 1 & 0 & 0 \end{array}$$

Cost : 2

$$\begin{array}{lccccc} \hline T & 1 & 1 & 1 & 0 & 0 \\ + & & & & & \\ x_2 & & & & & 1 \\ \hline & 1 & 1 & 1 & 0 & 1 \end{array}$$

Example: Say $|T| = 27 = \langle 11011 \rangle_2$ $\quad$ $T = \langle B_4\ B_3\ B_1\ B_0 \rangle$

$$\begin{array}{lccccc} & & & 1 & 1 & \\ \hline T & 1 & 1 & 0 & 1 & 1 \\ + & & & & & \\ x_1 & & & & & 1 \\ \hline & 1 & 1 & 1 & 0 & 0 \end{array}$$

Cost : 2

$$\begin{array}{lccccc} \hline T & 1 & 1 & 1 & 0 & 0 \\ + & & & & & \\ x_2 & & & & & 1 \\ \hline & 1 & 1 & 1 & 0 & 1 \end{array}$$

Cost : 0

Example: Say $|T| = 27 = < 11011 >_2$ $\quad\quad T = < B_4\ B_3\ B_1\ B_0 >$

$$
\begin{array}{lccccc}
 & & & 1 & 1 & \\
\hline
T & 1 & 1 & 0 & 1 & 1 \\
+ & & & & & \\
x_1 & & & & & 1 \\
\hline
 & 1 & 1 & 1 & 0 & 0
\end{array}
$$

Cost : 2

$$
\begin{array}{lccccc}
\hline
T & 1 & 1 & 1 & 0 & 0 \\
+ & & & & & \\
x_2 & & & & & 1 \\
\hline
 & 1 & 1 & 1 & 0 & 1
\end{array}
$$

Cost : 0

$$
\begin{array}{lccccc}
\hline
T & 1 & 1 & 1 & 0 & 1 \\
+ & & & & & \\
x_3 & & & & & 1 \\
\hline
\end{array}
$$

Example: Say $|T| = 27 = \langle 11011 \rangle_2$ $\quad$ $T = \langle B_4\ B_3\ B_1\ B_0 \rangle$

$$\begin{array}{lccccc} & & & 1 & 1 & \\ \hline T & 1 & 1 & 0 & 1 & 1 \\ + & & & & & \\ x_1 & & & & & 1 \\ \hline & 1 & 1 & 1 & 0 & 0 \end{array}$$

Cost : 2

$$\begin{array}{lccccc} \hline T & 1 & 1 & 1 & 0 & 0 \\ + & & & & & \\ x_2 & & & & & 1 \\ \hline & 1 & 1 & 1 & 0 & 1 \end{array}$$

Cost : 0

$$\begin{array}{lccccc} & & & & 1 & \\ \hline T & 1 & 1 & 1 & 0 & 1 \\ + & & & & & \\ x_3 & & & & & 1 \\ \hline & 1 & 1 & 1 & 1 & 0 \end{array}$$

Example: Say $|T| = 27 = \langle 11011 \rangle_2$ $\quad T = \langle B_4\, B_3\, B_1\, B_0 \rangle$

$$\begin{array}{lccccc}
 & & & 1 & 1 & \\
\hline
T & 1 & 1 & 0 & 1 & 1 \\
+ & & & & & \\
x_1 & & & & & 1 \\
\hline
 & 1 & 1 & 1 & 0 & 0
\end{array}$$

Cost : 2

$$\begin{array}{lccccc}
 & & & & & \\
\hline
T & 1 & 1 & 1 & 0 & 0 \\
+ & & & & & \\
x_2 & & & & & 1 \\
\hline
 & 1 & 1 & 1 & 0 & 1
\end{array}$$

Cost : 0

$$\begin{array}{lccccc}
 & & & & 1 & \\
\hline
T & 1 & 1 & 1 & 0 & 1 \\
+ & & & & & \\
x_3 & & & & & 1 \\
\hline
 & 1 & 1 & 1 & 1 & 0
\end{array}$$

Cost : 1

Example: Say $|T| = 27 = <11011>_2$ $T = <B_4\ B_3\ B_1\ B_0>$

$$\begin{array}{lccccc} & & 1 & 1 & & \\ T & 1 & 1 & 0 & 1 & 1 \\ + & & & & & \\ x_1 & & & & & 1 \\ \hline & 1 & 1 & 1 & 0 & 0 \end{array}$$

Cost : 2

$$\begin{array}{lccccc} T & 1 & 1 & 1 & 0 & 0 \\ + & & & & & \\ x_2 & & & & & 1 \\ \hline & 1 & 1 & 1 & 0 & 1 \end{array}$$

Cost : 0

$$\begin{array}{lccccc} & & & & 1 & \\ T & 1 & 1 & 1 & 0 & 1 \\ + & & & & & \\ x_3 & & & & & 1 \\ \hline & 1 & 1 & 1 & 1 & 0 \end{array}$$

Cost : 1

$$\begin{array}{lccccc} T & 1 & 1 & 1 & 1 & 0 \\ + & & & & & \\ x_4 & & & & & 1 \\ \hline & & & & & \end{array}$$

Example: Say $|T| = 27 = \langle 11011 \rangle_2$ $\quad$ $T = \langle B_4\ B_3\ B_1\ B_0 \rangle$

$$\begin{array}{l|ccccc} & & & 1 & 1 & \\ \hline T & 1 & 1 & 0 & 1 & 1 \\ + & & & & & \\ x_1 & & & & & 1 \\ \hline & 1 & 1 & 1 & 0 & 0 \end{array}$$

Cost : 2

$$\begin{array}{l|ccccc} & & & & & \\ \hline T & 1 & 1 & 1 & 0 & 0 \\ + & & & & & \\ x_2 & & & & & 1 \\ \hline & 1 & 1 & 1 & 0 & 1 \end{array}$$

Cost : 0

$$\begin{array}{l|ccccc} & & & & 1 & \\ \hline T & 1 & 1 & 1 & 0 & 1 \\ + & & & & & \\ x_3 & & & & & 1 \\ \hline & 1 & 1 & 1 & 1 & 0 \end{array}$$

Cost : 1

$$\begin{array}{l|ccccc} & & & & & \\ \hline T & 1 & 1 & 1 & 1 & 0 \\ + & & & & & \\ x_4 & & & & & 1 \\ \hline & 1 & 1 & 1 & 1 & 1 \end{array}$$

Example: Say $|T| = 27 = < 11011 >_2$ $\qquad$ $T = < B_4\ B_3\ B_1\ B_0 >$

$$
\begin{array}{c|ccccc}
 & & & 1 & 1 & \\
\hline
T & 1 & 1 & 0 & 1 & 1 \\
+ & & & & & \\
x_1 & & & & & 1 \\
\hline
 & 1 & 1 & 1 & 0 & 0
\end{array}
$$

Cost : 2

$$
\begin{array}{c|ccccc}
T & 1 & 1 & 1 & 0 & 0 \\
+ & & & & & \\
x_2 & & & & & 1 \\
\hline
 & 1 & 1 & 1 & 0 & 1
\end{array}
$$

Cost : 0

$$
\begin{array}{c|ccccc}
 & & & & 1 & \\
\hline
T & 1 & 1 & 1 & 0 & 1 \\
+ & & & & & \\
x_3 & & & & & 1 \\
\hline
 & 1 & 1 & 1 & 1 & 0
\end{array}
$$

Cost : 1

$$
\begin{array}{c|ccccc}
T & 1 & 1 & 1 & 1 & 0 \\
+ & & & & & \\
x_4 & & & & & 1 \\
\hline
 & 1 & 1 & 1 & 1 & 1
\end{array}
$$

Cost : 0

Example: Say $|T| = 27 = \langle 11011 \rangle_2$ $T = \langle B_4\ B_3\ B_1\ B_0 \rangle$

$$\begin{array}{lccccc} & & & 1 & 1 & \\ T & 1 & 1 & 0 & 1 & 1 \\ + & & & & & \\ x_1 & & & & & 1 \\ \hline & 1 & 1 & 1 & 0 & 0 \end{array}$$

Cost : 2

$$\begin{array}{lccccc} T & 1 & 1 & 1 & 0 & 0 \\ + & & & & & \\ x_2 & & & & & 1 \\ \hline & 1 & 1 & 1 & 0 & 1 \end{array}$$

Cost : 0

$$\begin{array}{lccccc} & & & & 1 & \\ T & 1 & 1 & 1 & 0 & 1 \\ + & & & & & \\ x_3 & & & & & 1 \\ \hline & 1 & 1 & 1 & 1 & 0 \end{array}$$

Cost : 1

$$\begin{array}{lccccc} T & 1 & 1 & 1 & 1 & 0 \\ + & & & & & \\ x_4 & & & & & 1 \\ \hline & 1 & 1 & 1 & 1 & 1 \end{array}$$

Cost : 0

$$\begin{array}{lccccc} T & 1 & 1 & 1 & 1 & 1 \\ + & & & & & \\ x_5 & & & & & 1 \\ \hline & & & & & \end{array}$$

Example: Say $|T| = 27 = \langle 11011 \rangle_2$ $\quad$ $T = \langle B_4\, B_3\, B_1\, B_0 \rangle$

$$\begin{array}{llllll} & & & 1 & 1 & \\ T & 1 & 1 & 0 & 1 & 1 \\ + & & & & & \\ x_1 & & & & & 1 \\ \hline & 1 & 1 & 1 & 0 & 0 \end{array}$$

Cost : 2

$$\begin{array}{llllll} & & & & & \\ T & 1 & 1 & 1 & 0 & 0 \\ + & & & & & \\ x_2 & & & & & 1 \\ \hline & 1 & 1 & 1 & 0 & 1 \end{array}$$

Cost : 0

$$\begin{array}{llllll} & & & & 1 & \\ T & 1 & 1 & 1 & 0 & 1 \\ + & & & & & \\ x_3 & & & & & 1 \\ \hline & 1 & 1 & 1 & 1 & 0 \end{array}$$

Cost : 1

$$\begin{array}{llllll} & & & & & \\ T & 1 & 1 & 1 & 1 & 0 \\ + & & & & & \\ x_4 & & & & & 1 \\ \hline & 1 & 1 & 1 & 1 & 1 \end{array}$$

Cost : 0

$$\begin{array}{lllllll} & 1 & 1 & 1 & 1 & 1 & \\ T & & 1 & 1 & 1 & 1 & 1 \\ + & & & & & & \\ x_5 & & & & & & 1 \\ \hline & 1 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Example: Say $|T| = 27 = \langle 11011 \rangle_2$ $\quad T = \langle B_4\ B_3\ B_1\ B_0 \rangle$

$$\begin{array}{l|rrrrrr} & & & & 1 & 1 & \\ T & & 1 & 1 & 0 & 1 & 1 \\ + & & & & & & \\ x_1 & & & & & & 1 \\ \hline & & 1 & 1 & 1 & 0 & 0 \end{array}$$

Cost : 2

$$\begin{array}{l|rrrrrr} T & & 1 & 1 & 1 & 0 & 0 \\ + & & & & & & \\ x_2 & & & & & & 1 \\ \hline & & 1 & 1 & 1 & 0 & 1 \end{array}$$

Cost : 0

$$\begin{array}{l|rrrrrr} & & & & & 1 & \\ T & & 1 & 1 & 1 & 0 & 1 \\ + & & & & & & \\ x_3 & & & & & & 1 \\ \hline & & 1 & 1 & 1 & 1 & 0 \end{array}$$

Cost : 1

$$\begin{array}{l|rrrrrr} T & & 1 & 1 & 1 & 1 & 0 \\ + & & & & & & \\ x_4 & & & & & & 1 \\ \hline & & 1 & 1 & 1 & 1 & 1 \end{array}$$

Cost : 0

$$\begin{array}{l|rrrrrr} & 1 & 1 & 1 & 1 & 1 & \\ T & & 1 & 1 & 1 & 1 & 1 \\ + & & & & & & \\ x_5 & & & & & & 1 \\ \hline & 1 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Cost : 5

Example: Say $|T| = 27 = \langle 11011 \rangle_2$ $\quad T = \langle B_4\ B_3\ B_1\ B_0 \rangle$

	x_1	x_2	x_3	x_4	x_5
Carries	1 1		1		1 1 1 1 1
T	1 1 0 1 1	1 1 1 0 0	1 1 1 0 1	1 1 1 1 0	1 1 1 1 1
$+\ x_i$	1	1	1	1	1
Result	1 1 1 0 0	1 1 1 0 1	1 1 1 1 0	1 1 1 1 1	1 0 0 0 0 0
Cost	2	0	1	0	5

- Total for 5 insertions: 2 + 0 + 1 + 0 + 5 = 8 key-comparisons (not 5 x 5).

Example: Say $|T| = 27 = \langle 11011 \rangle_2$ $\quad T = \langle B_4\ B_3\ B_1\ B_0 \rangle$

$$\begin{array}{lccccc} & & & 1 & 1 & \\ \hline T & 1 & 1 & 0 & 1 & 1 \\ + & & & & & \\ x_1 & & & & & 1 \\ \hline & 1 & 1 & 1 & 0 & 0 \end{array}$$

Cost : 2

$$\begin{array}{lccccc} & & & & & \\ \hline T & 1 & 1 & 1 & 0 & 0 \\ + & & & & & \\ x_2 & & & & & 1 \\ \hline & 1 & 1 & 1 & 0 & 1 \end{array}$$

Cost : 0

$$\begin{array}{lccccc} & & & & 1 & \\ \hline T & 1 & 1 & 1 & 0 & 1 \\ + & & & & & \\ x_3 & & & & & 1 \\ \hline & 1 & 1 & 1 & 1 & 0 \end{array}$$

Cost : 1

$$\begin{array}{lccccc} & & & & & \\ \hline T & 1 & 1 & 1 & 1 & 0 \\ + & & & & & \\ x_4 & & & & & 1 \\ \hline & 1 & 1 & 1 & 1 & 1 \end{array}$$

Cost : 0

$$\begin{array}{lcccccc} & 1 & 1 & 1 & 1 & 1 & \\ \hline T & & 1 & 1 & 1 & 1 & 1 \\ + & & & & & & \\ x_5 & & & & & & 1 \\ \hline & 1 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Cost : 5

- Total for 5 insertions: 2 + 0 + 1 + 0 + 5 = 8 key-comparisons (not 5 x 5).
- Initially: T has 27 - α(27) = 27 – 4 = 23 edges

Example: Say $|T| = 27 = <11011>_2$ $\quad T = <B_4\ B_3\ B_1\ B_0>$

$$\begin{array}{lcccccc} & & & & 1 & 1 & \\ \hline T & & 1 & 1 & 0 & 1 & 1 \\ + & & & & & & \\ x_1 & & & & & & 1 \\ \hline & & 1 & 1 & 1 & 0 & 0 \end{array}$$

Cost : 2

$$\begin{array}{lcccccc} & & & & & & \\ \hline T & & 1 & 1 & 1 & 0 & 0 \\ + & & & & & & \\ x_2 & & & & & & 1 \\ \hline & & 1 & 1 & 1 & 0 & 1 \end{array}$$

Cost : 0

$$\begin{array}{lcccccc} & & & & & 1 & \\ \hline T & & 1 & 1 & 1 & 0 & 1 \\ + & & & & & & \\ x_3 & & & & & & 1 \\ \hline & & 1 & 1 & 1 & 1 & 0 \end{array}$$

Cost : 1

$$\begin{array}{lcccccc} & & & & & & \\ \hline T & & 1 & 1 & 1 & 1 & 0 \\ + & & & & & & \\ x_4 & & & & & & 1 \\ \hline & & 1 & 1 & 1 & 1 & 1 \end{array}$$

Cost : 0

$$\begin{array}{lcccccc} & 1 & 1 & 1 & 1 & 1 & \\ \hline T & & 1 & 1 & 1 & 1 & 1 \\ + & & & & & & \\ x_5 & & & & & & 1 \\ \hline & 1 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Cost : 5

- Total for 5 insertions: 2 + 0 + 1 + 0 + 5 = 8 key-comparisons (not 5 x 5).
- Initially: T has 27 - $\alpha(27)$ = 27 – 4 = 23 edges
- After 5 insertions: T has 32 - $\alpha(32)$ = 32 – 1 = 31 edges

Example: Say $|T| = 27 = \langle 11011 \rangle_2$ $\quad T = \langle B_4\ B_3\ B_1\ B_0 \rangle$

carries: 1 1		carries: 1		carries: 1 1 1 1 1
T: 1 1 0 1 1	T: 1 1 1 0 0	T: 1 1 1 0 1	T: 1 1 1 1 0	T: 1 1 1 1 1
+ x_1: 1	+ x_2: 1	+ x_3: 1	+ x_4: 1	+ x_5: 1
1 1 1 0 0	1 1 1 0 1	1 1 1 1 0	1 1 1 1 1	1 0 0 0 0 0
Cost : 2	Cost : 0	Cost : 1	Cost : 0	Cost : 5

- Total for 5 insertions: 2 + 0 + 1 + 0 + 5 = 8 key-comparisons (not 5 x 5).
- Initially: T has 27 - α(27) = 27 – 4 = 23 edges
- After 5 insertions: T has 32 - α(32) = 32 – 1 = 31 edges
- The 5 insertions added: 31-23 = 8 new edges.

Example: Say $|T| = 27 = \langle 11011 \rangle_2$ $\qquad$ $T = \langle B_4\ B_3\ B_1\ B_0 \rangle$

	$T + x_1$	$T + x_2$	$T + x_3$	$T + x_4$	$T + x_5$
Carries	1 1		1		1 1 1 1 1
T	1 1 0 1 1	1 1 1 0 0	1 1 1 0 1	1 1 1 1 0	1 1 1 1 1
$+\ x_i$	1	1	1	1	1
Result	1 1 1 0 0	1 1 1 0 1	1 1 1 1 0	1 1 1 1 1	1 0 0 0 0 0
Cost	2	0	1	0	5

- Total for 5 insertions: 2 + 0 + 1 + 0 + 5 = 8 key-comparisons (not 5 x 5).
- Initially: T has 27 - α(27) = 27 – 4 = 23 edges
- After 5 insertions: T has 32 - α(32) = 32 – 1 = 31 edges
- The 5 insertions added: 31-23 = 8 new edges.
- 8 new edges = 8 key-comparisons

Total Cost of k successive inserts into a Binomial Heap with n nodes ?

Total Cost of k successive inserts
into a Binomial Heap with n nodes ?

total cost is at most $O(k \log (n + k))$ key-comparisons

Total Cost of k successive inserts into a Binomial Heap with n nodes ?

Claim: If $k > \log_2 n$, total cost is at most key-comparisons

Total Cost of k successive inserts
into a Binomial Heap with n nodes ?

Claim: If $k > \log_2 n$, total cost is at most $2k$ key-comparisons

Total Cost of k successive inserts into a Binomial Heap with n nodes ?

Claim: If $k > \log_2 n$, total cost is at most $2k$ key-comparisons

Proof: Do A2 – Q6 ☺

Total Cost of k successive inserts
into a Binomial Heap with n nodes ?

Claim: If $k > \log_2 n$, total cost is at most $2k$ key-comparisons

Proof: Do A2 – Q6 ☺

⇒ Average cost per insert is ≤ 2 key-comparisons !