(2–4) Tree
(Balanced Search Tree)

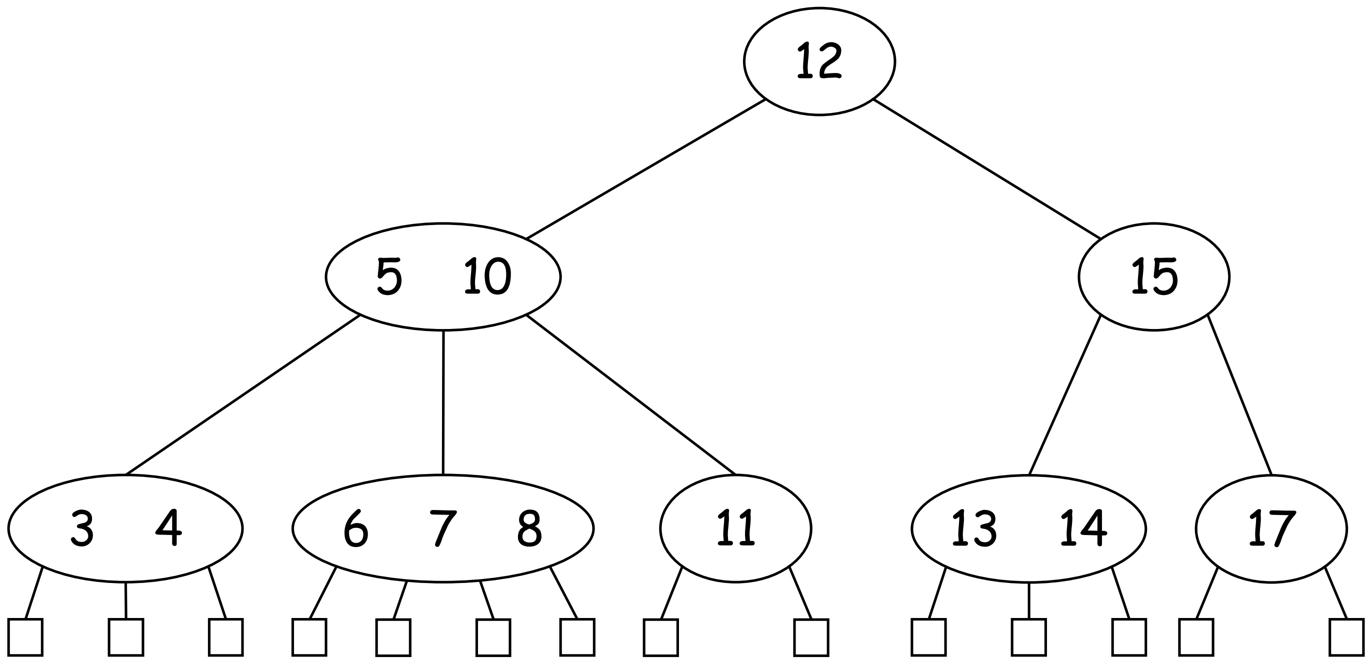
(2, 4) Trees

- **Size Property**: Every node has at most four children.
- **Depth Property**: All the external nodes have the same depth.
- See Figure 3.18.

***Theorem* 3.4**
The height of a (2, 4) tree storing n items is $\Theta(\lg n)$.

Proof.
The number of external nodes = $n + 1$. (by *Theorem* 3.3)
By the size property, $n + 1 \leq 4^h$.
By the depth property, $2^h \leq n + 1$.
Therefore, $h \leq \lg(n + 1)$ and $\lg(n + 1) \leq 2h$.

(2, 4) Trees



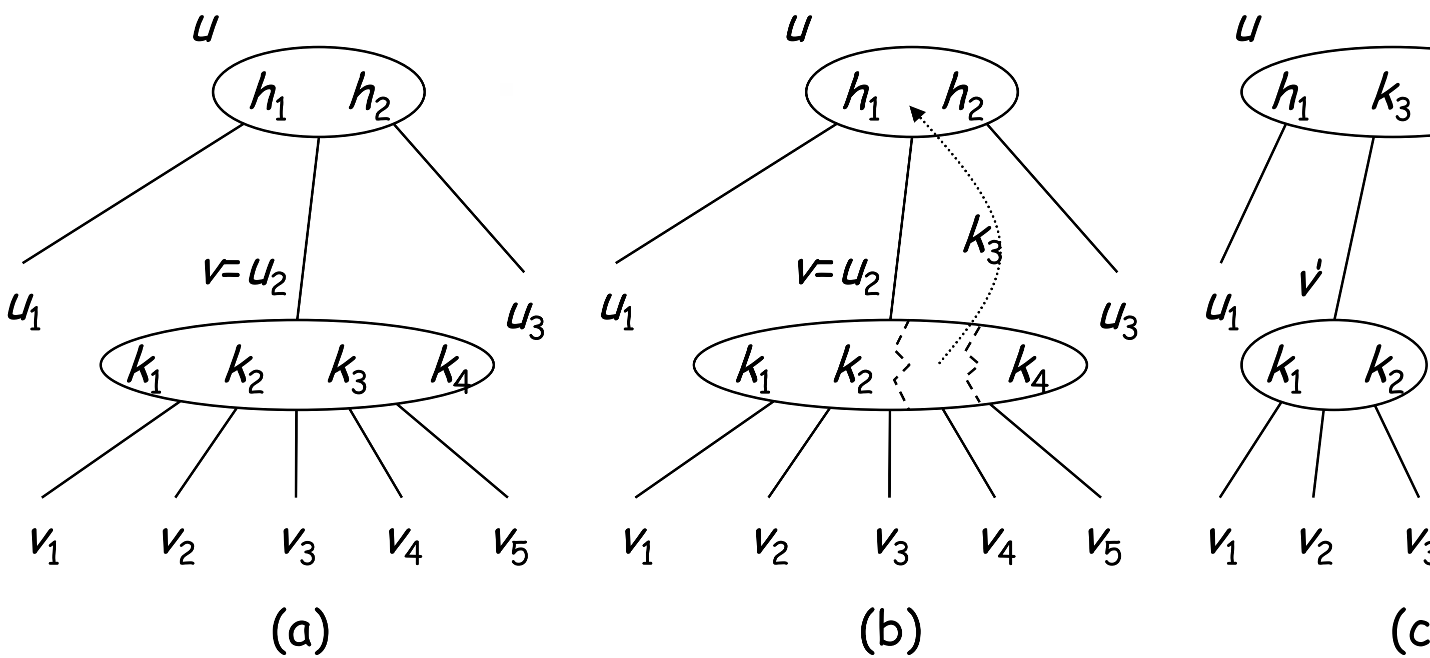
Insertion

- To insert a new item (k, x) into a (2, 4) tree T:
  - Search for key k in T and arrive at an external node z.
  - Insert the new item into the parent v of z and add a new external-node child w to v on the left of z. That is, we add (k, x, w) to $D(v)$.
  - Fix up any violation of size property (overflow).
- To remedy the overflow at v, we perform a split operation:

  $v_1, \ldots, v_5$: the children of v

  $k_1, \ldots, k_4$: the keys stored at v (See Figure 3.19.)
  - Replace v with two nodes v' and v'', where
    - v' is a 3-node with children v_1, v_2, v_3 storing keys k_1 and k_2.
    - v'' is a 2-node with children v_4, v_5 storing key k_4.

Insertion



- A node split
  - (a) Overflow at a 5-node v
  - (b) The third key of v inserted into the parent u of v
  - (c) Node v replaced with a 3-node v' and a 2-node v''

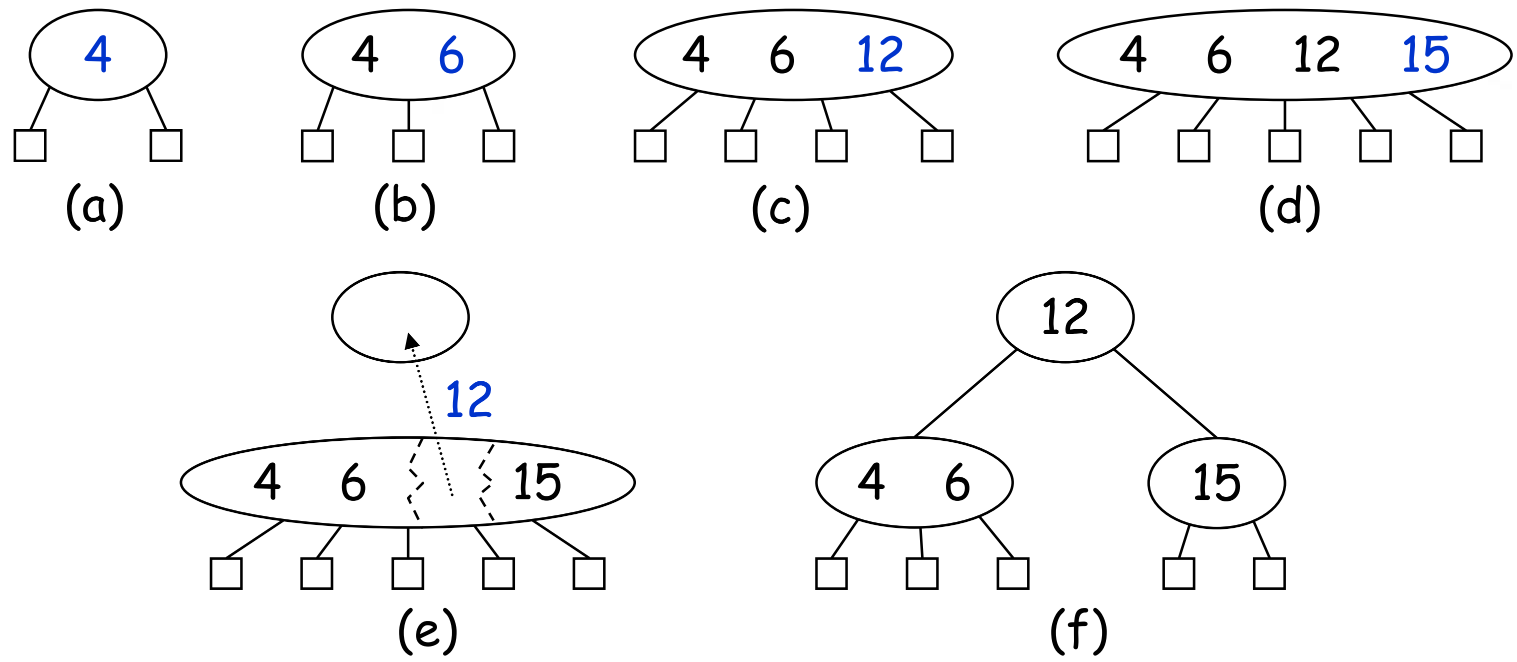
Insertion

- If v was the root of T, create a new root node u; else, let u be the parent of v.
- Insert key k_3 into u and make v' and v'' children of u, so that if v was child i of u, then v' and v'' become children i and $i + 1$ of u, respectively.

- See Figure 3.20.

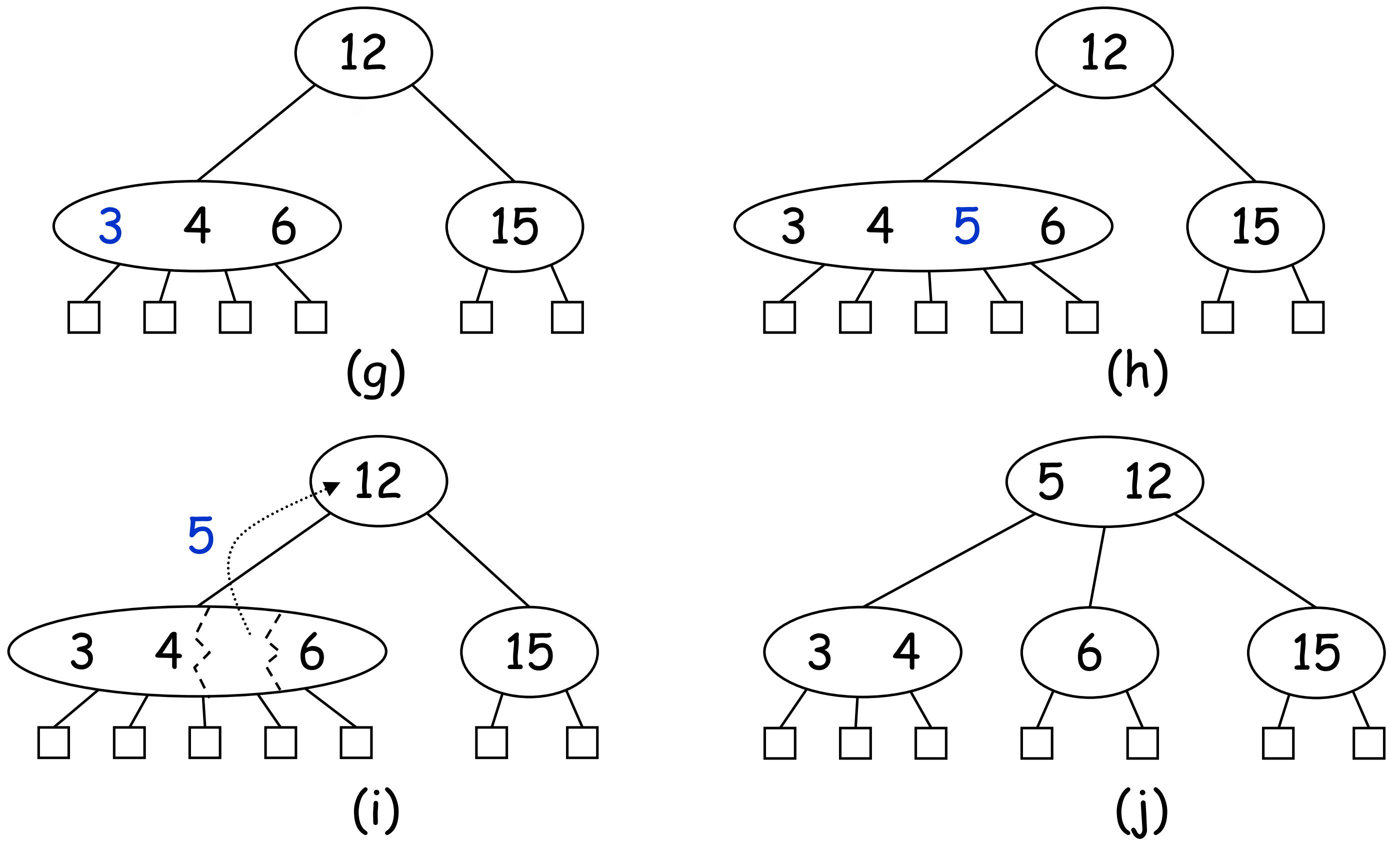
- A split operation either eliminates the overflow or propagates it into the parent of the current node.
  - The total time to perform an insertion in a (2, 4) tree is $O(\lg n)$.
- See Figure 3.21.

Insertion



- *A sequence of insertions into a (2, 4) tree*
  - (a) (b) (c) Initial tree with one item, Insertion of 6, 12
  - (d) Insertion of 15, which causes an overflow
  - (e) Split, which causes the creation of a new root node
  - (f) After the split

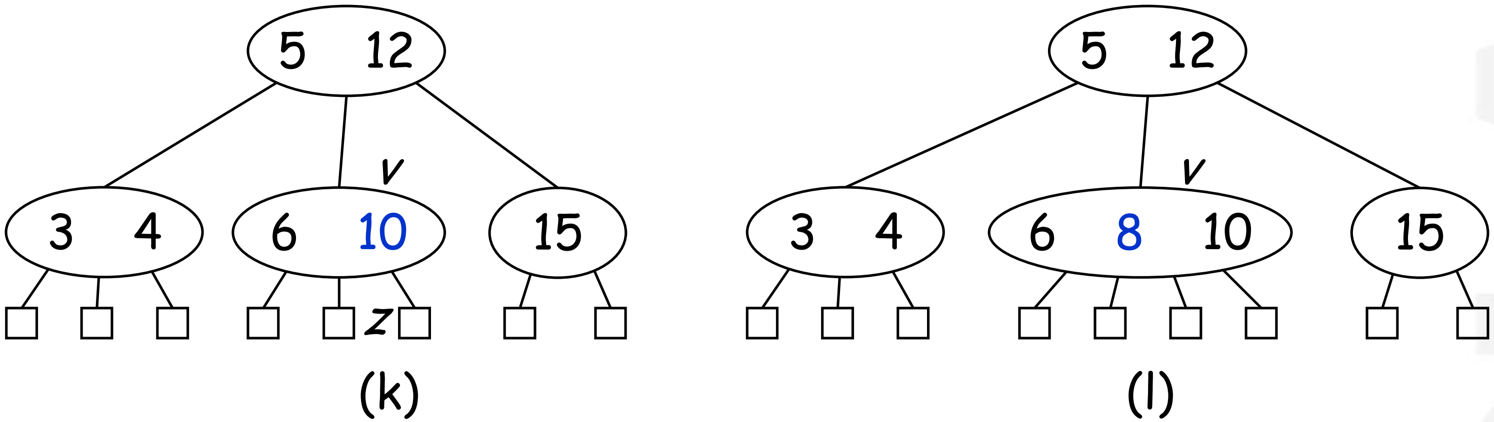
Insertion



(g) (h) Insertion of 3, 5

(i) (j) Split, After the split

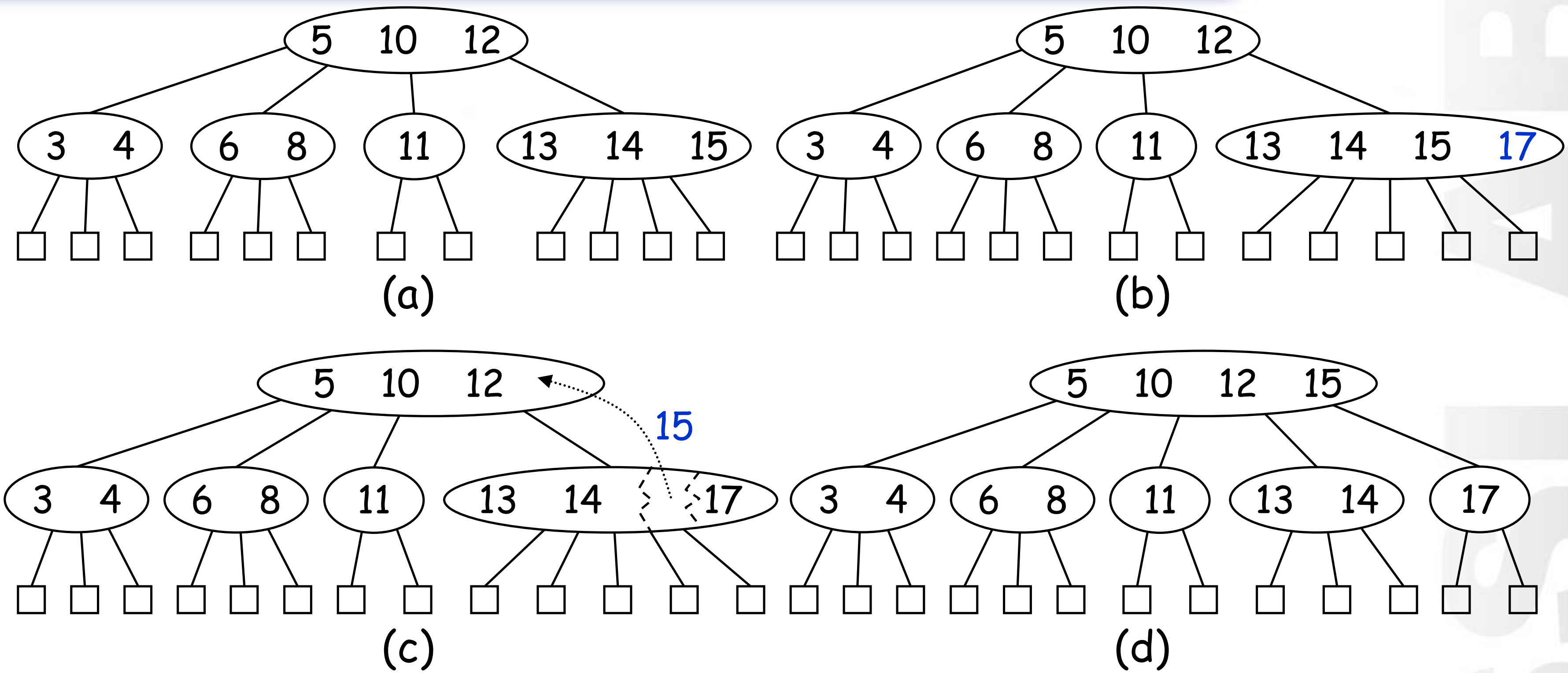
Insertion



(k) Insertion of 10

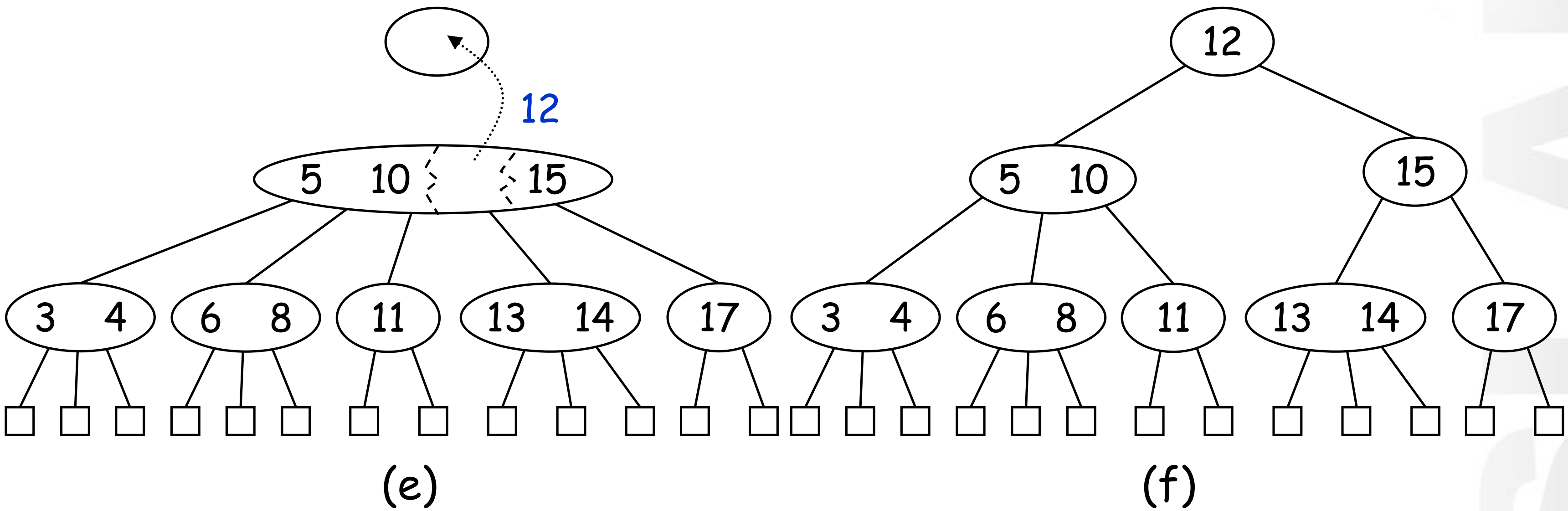
(l) Insertion of 8

Insertion



- A insertion in a (2, 4) tree that causes a cascading split
  - (a) Before the insertion
  - (b) Insertion of 17, causing a overflow
  - (c) A split
  - (d) After the split a new overflow occurs

Insertion



(e) Another split, creating a new root node
(d) Final tree

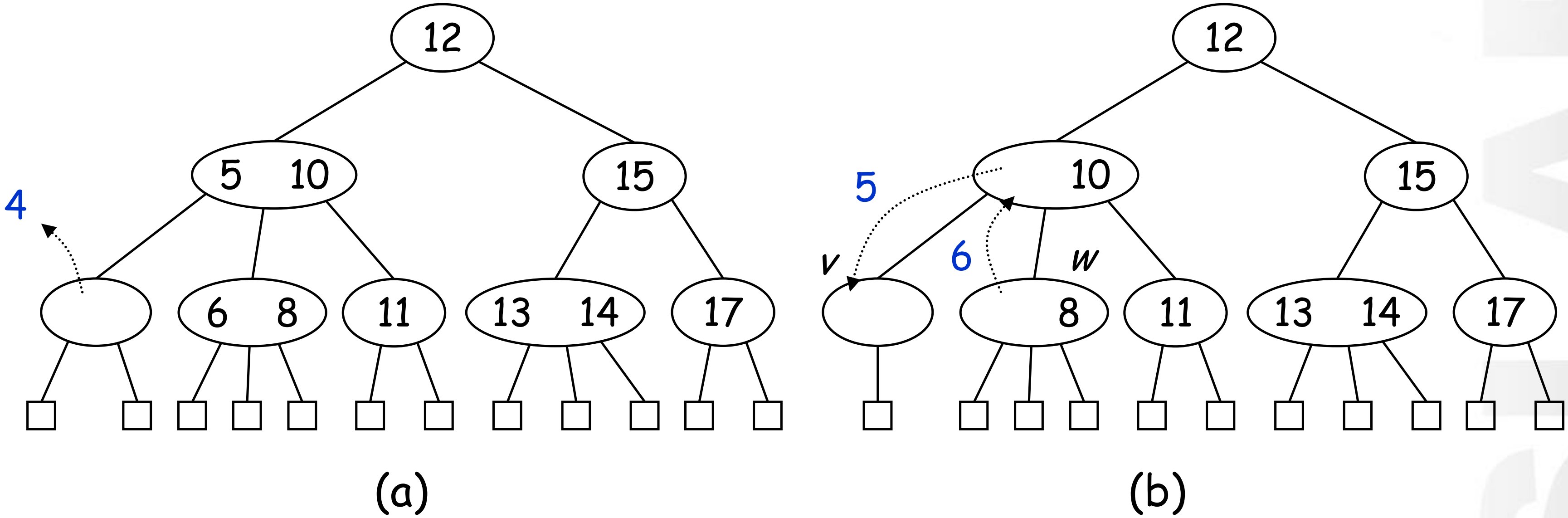
Removal

- If the item with key k that we wish to remove is stored in the ith item (k_i, x_i) at a node z that has only internal-node children. We swap the item (k_i, x_i) with an appropriate item that is stored at a node v with external-node children:
  - We find the right-most internal node v in the subtree rooted at the ith child of z, noting that the children of node v are all external nodes.
  - We swap the item (k_i, x_i) at z with the last item of v.
- Once we ensure that the item to remove is stored at a node v with only external-node children, we simply remove the item from v and remove the ith external node of v.
- Then, we may have to remedy an underflow at node v.

Removal

- To remedy an underflow at node v, we check whether an immediate sibling w of v is a 3-node or a 4-node:
  - **Transfer** operation when such a w is found: Move a child of w to v, a key of w to the parent u of v and w, and a key of y to v.
  - **Fusion** operation when v has only one sibling, or if both immediate siblings of v are 2-nodes: Merge v with a sibling, creating a new node v', and move a key from the parent u of v to v'.
- See Figure 3.22 and 3.23.
- *A fusion operation either eliminates the underflow or propagates it into the parent of the current node.*
  - Removal in a (2, 4) tree takes $O(\lg n)$ time.

Removal

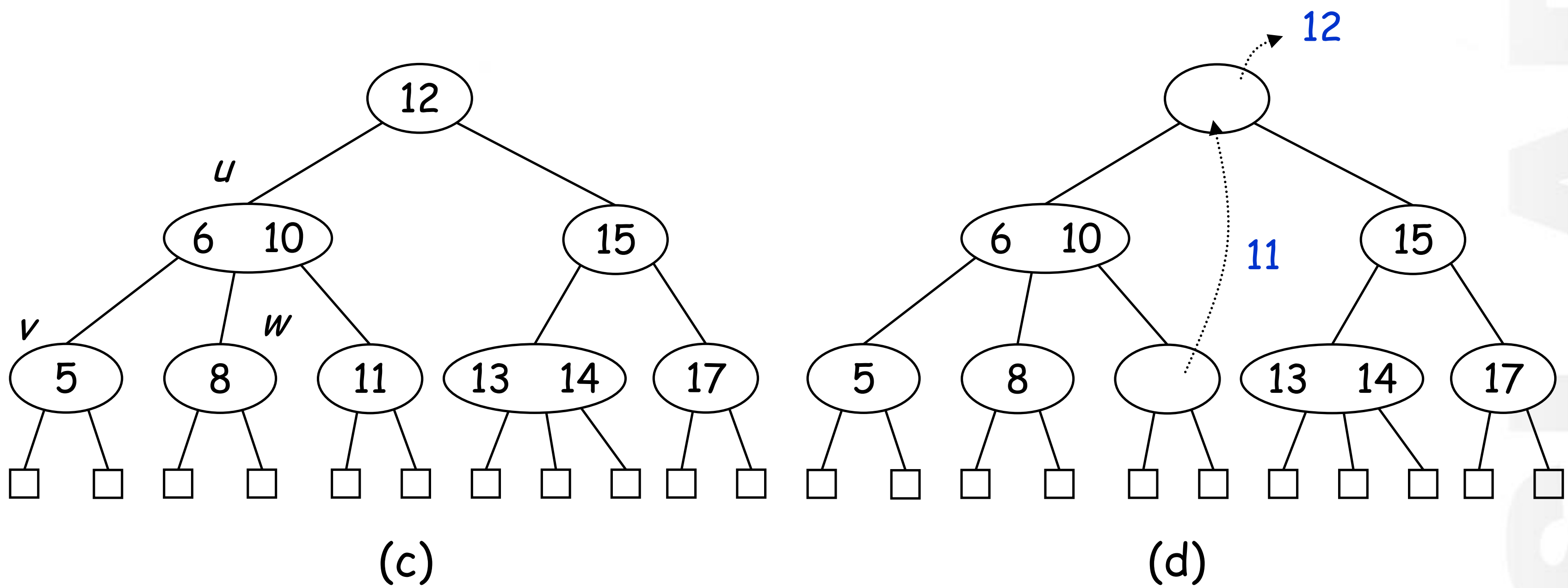


- *A sequence of removals from a (2, 4) tree*

  (a) Removal of 4, causing an underflow

  (b) A transfer operations

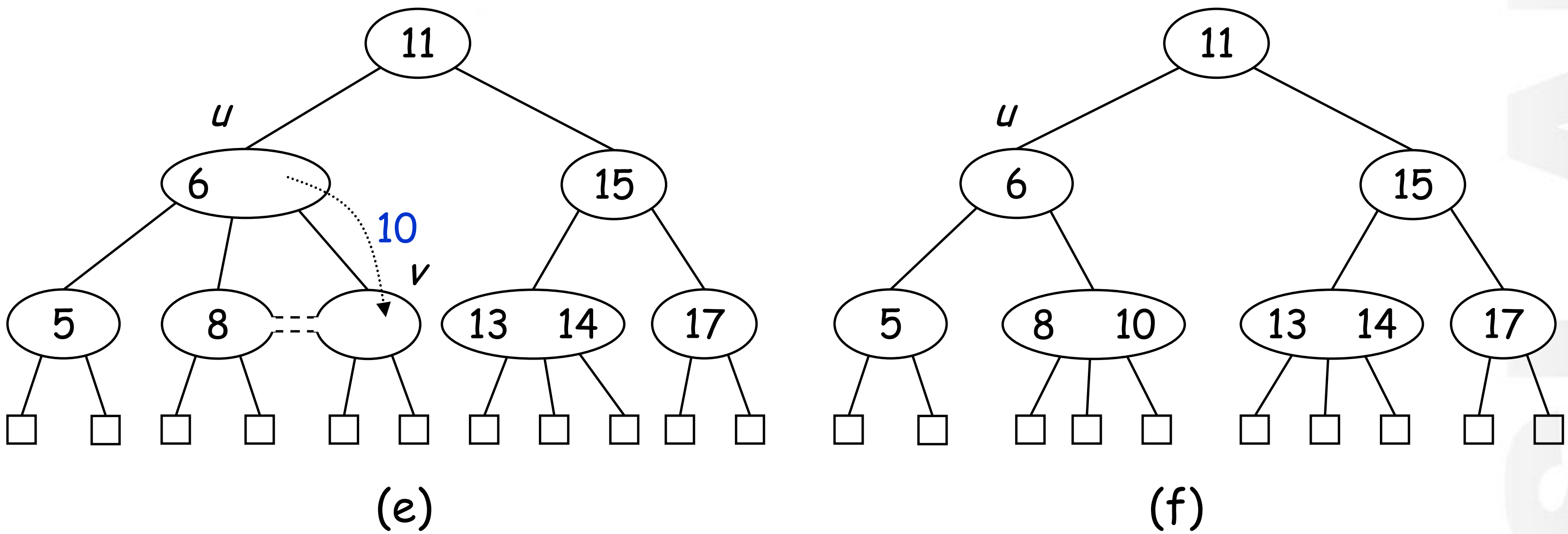
Removal



(c) After the transfer operation

(d) Removal of 12, causing an underflow

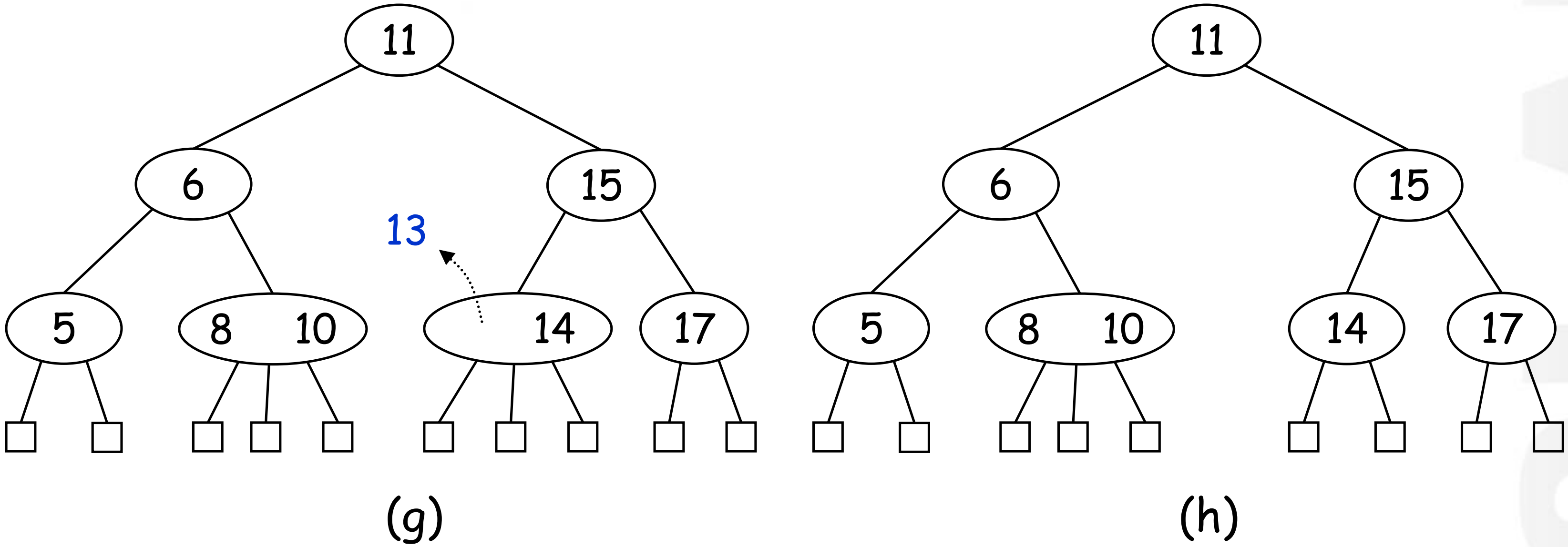
Removal



(e) A fusion operation

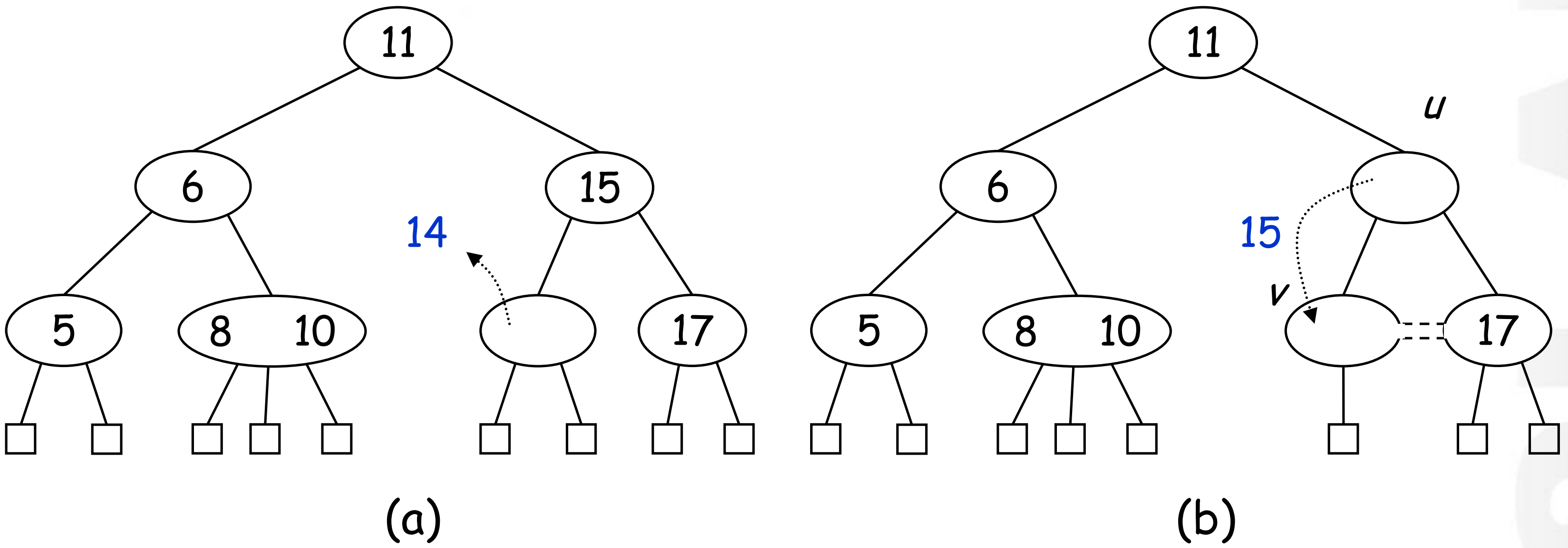
(f) After the fusion operation

Removal



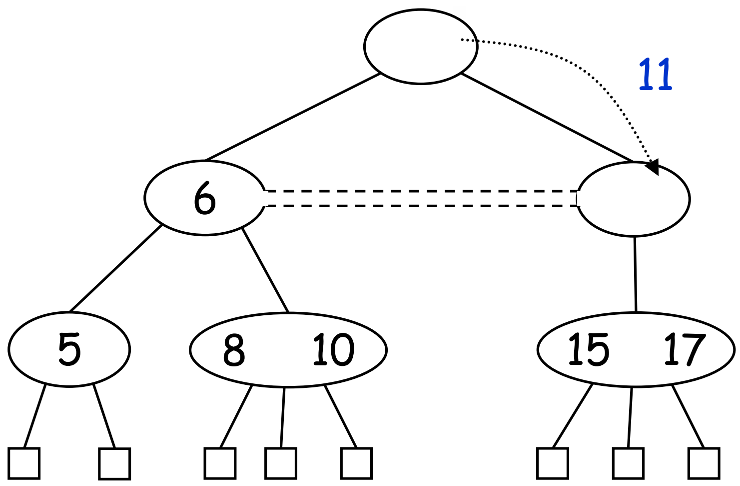
(g) Removal of 13
(h) After removing 13

Removal

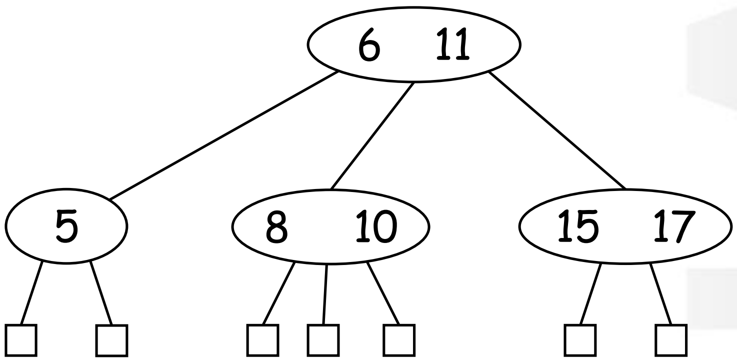


- *A propagating sequence of fusions in a (2, 4) tree*
  - (a) Removal of 14, which causes an underflow
  - (b) Fusion, which causes another underflow

Removal



(c)



(d)

(c) Second fusion operation, which causes the root to be removed

(d) Final tree