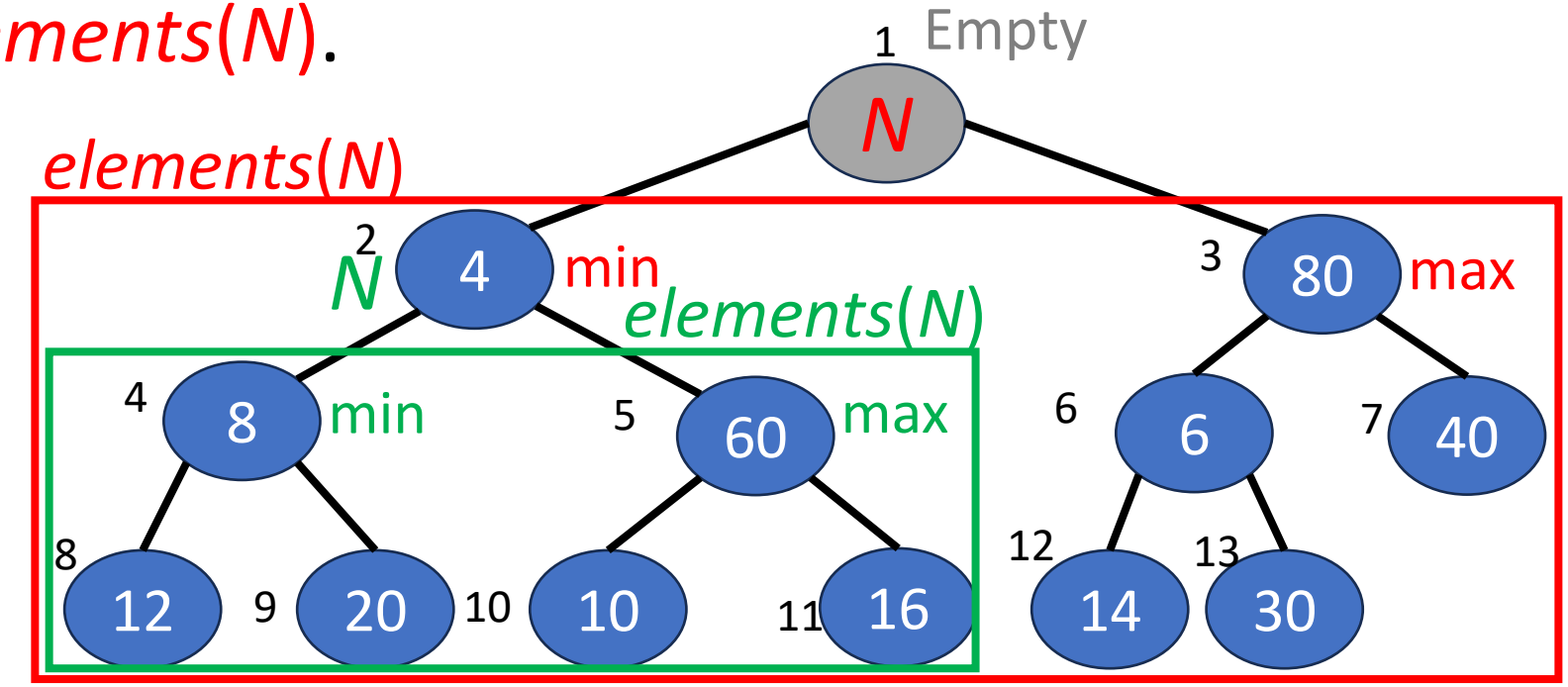
Min-Max Heaps

Ch. 9

Symmetric min-max heap (SMMH)

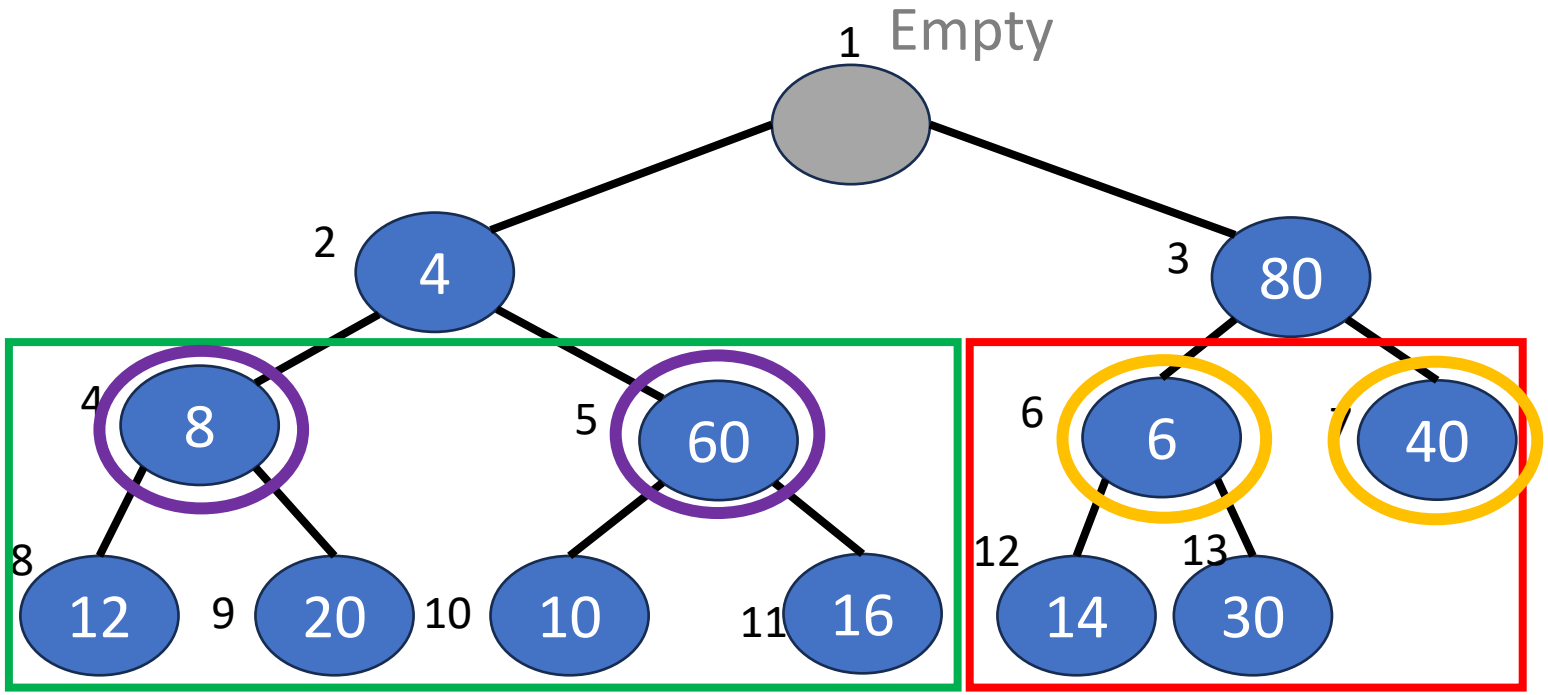
A complete binary tree empty or satisfying these properties:

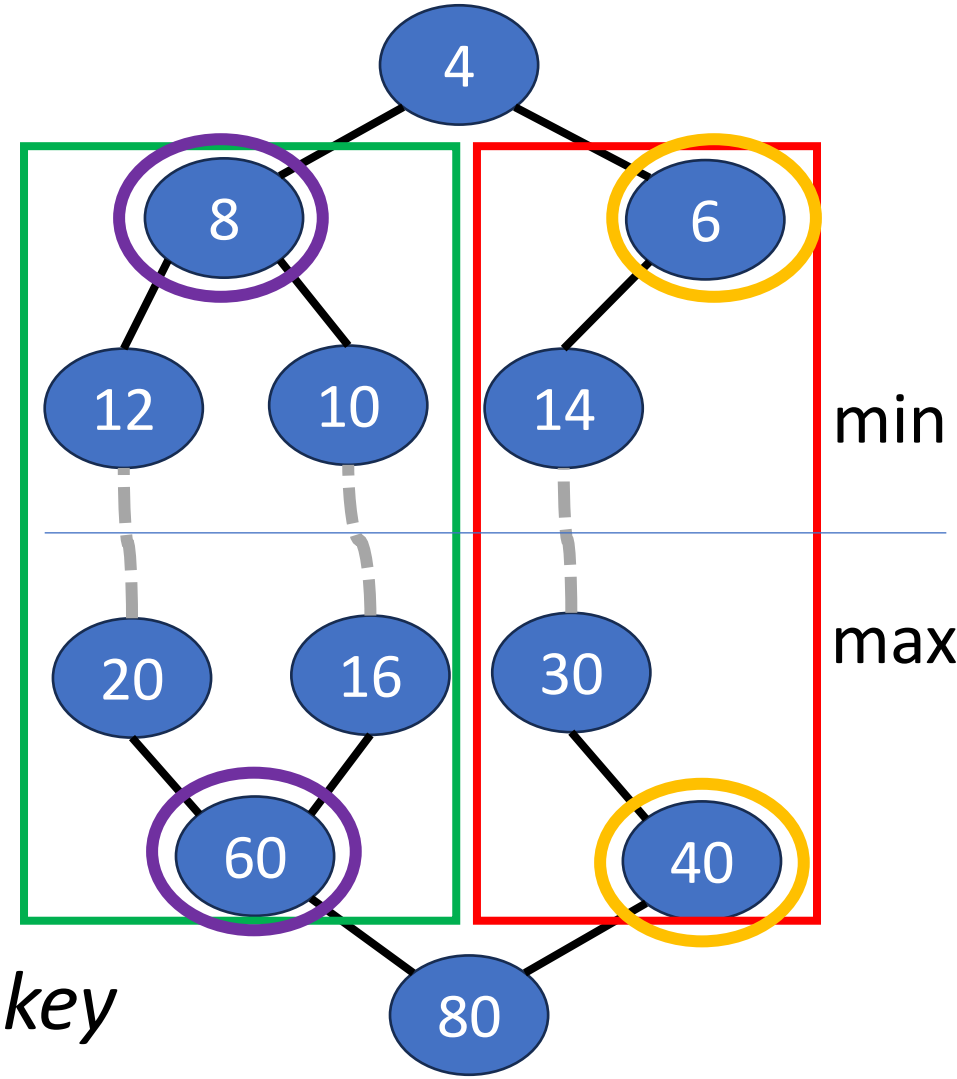
1. The root contains no element.
2. Given any node *N* in the heap:
   - The left child of *N* has the minimum element in *elements*(*N*).
   - The right child of *N* (if any) has the maximum element in *elements*(*N*).



Properties of SMMH

- What is the relationship of siblings?

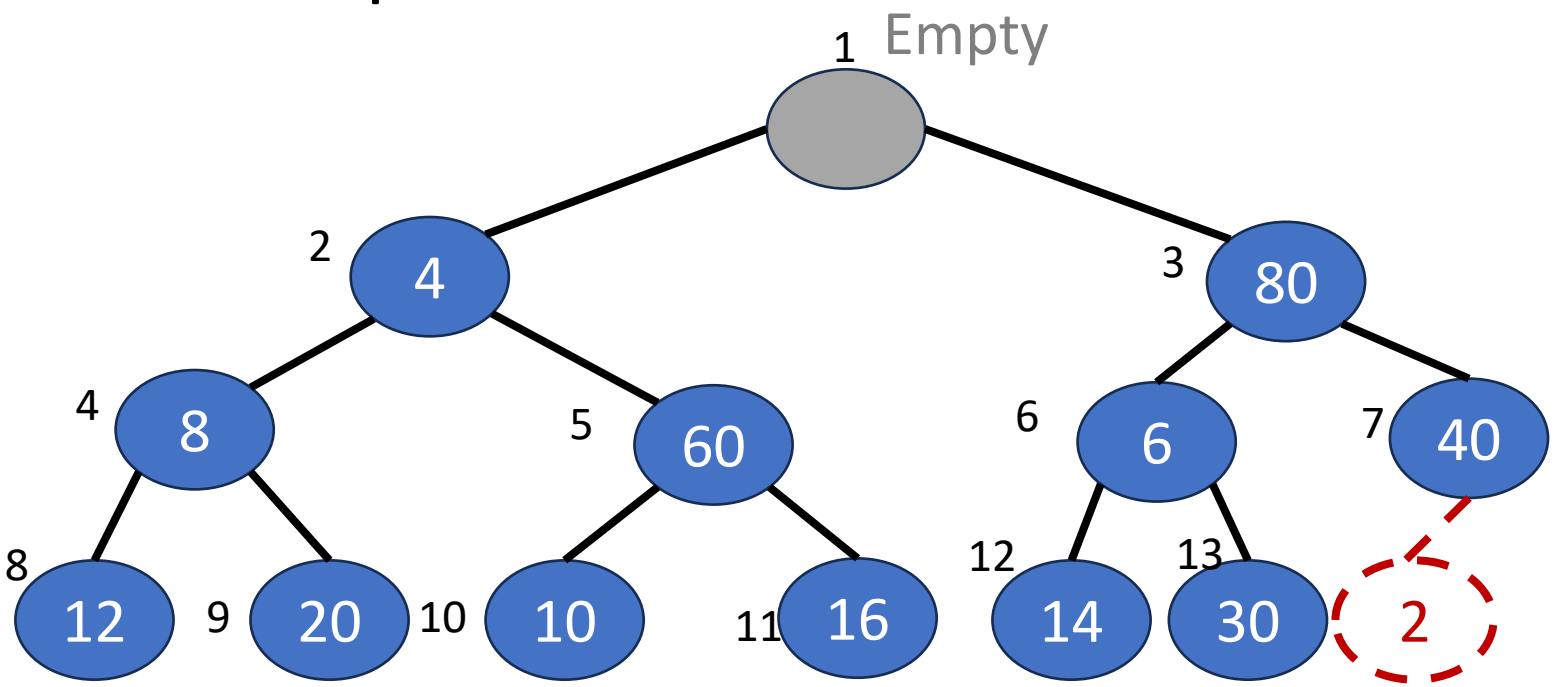


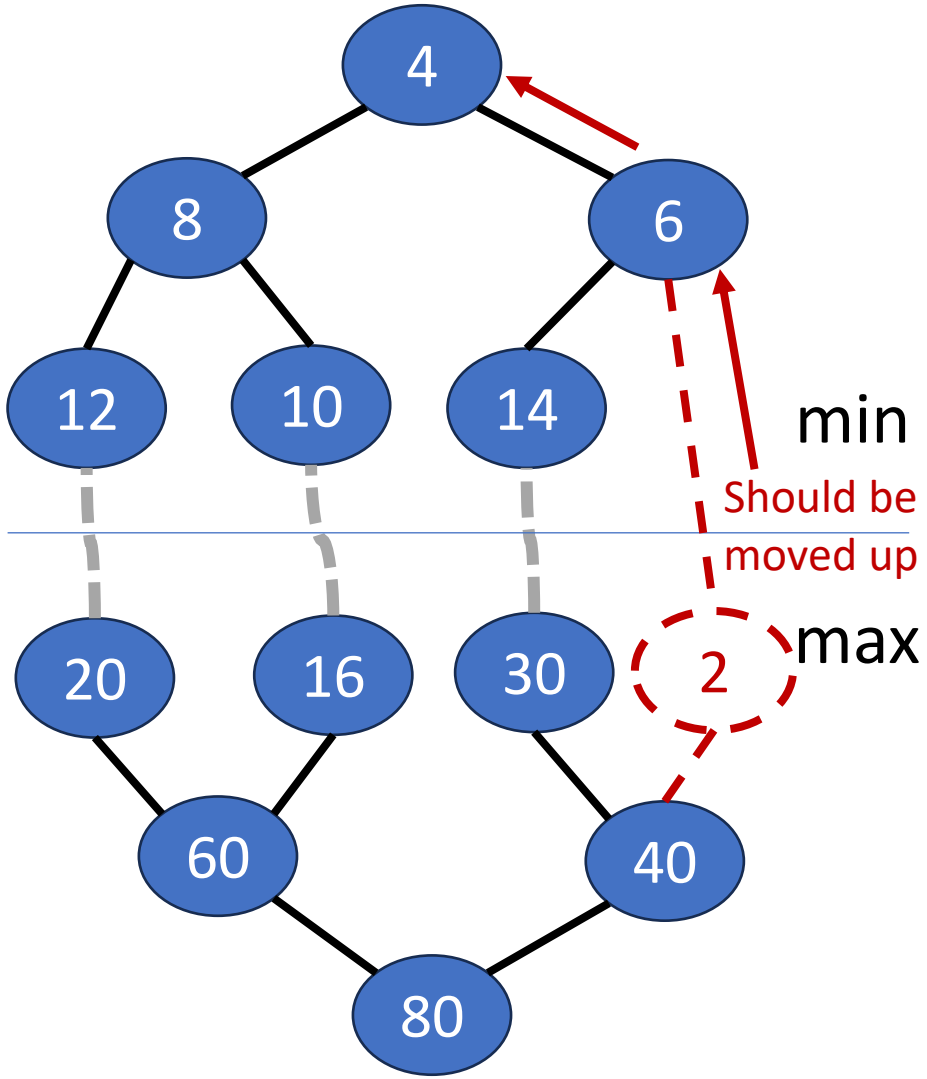


- For any node *N* with grandparent *GP*,
  GP->leftChild.key ≤ *N.key* ≤ *GP->rightChild.key*

Operation: Insertion

- Example: Insert 2

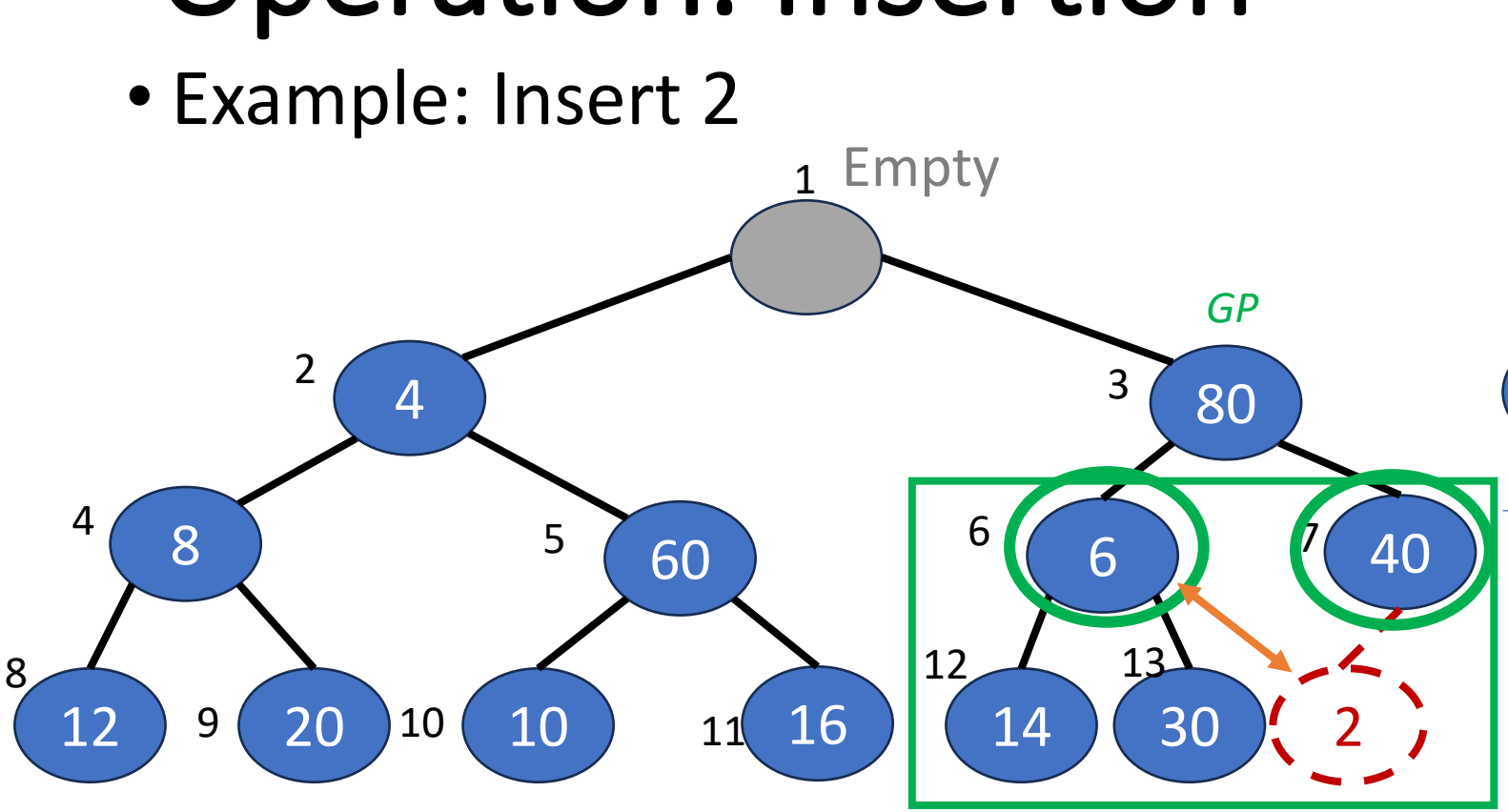


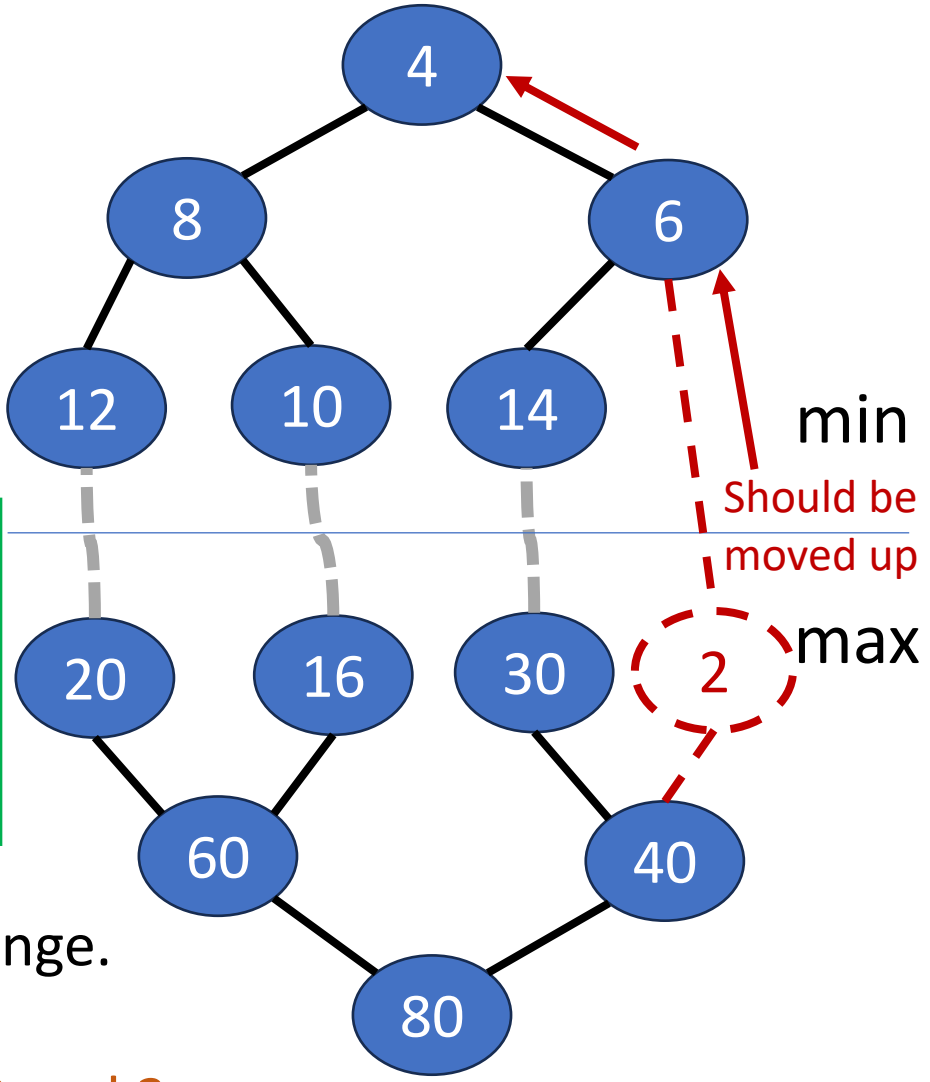


Step 1: Insert the new element after the last position.

Operation: Insertion

- Example: Insert 2





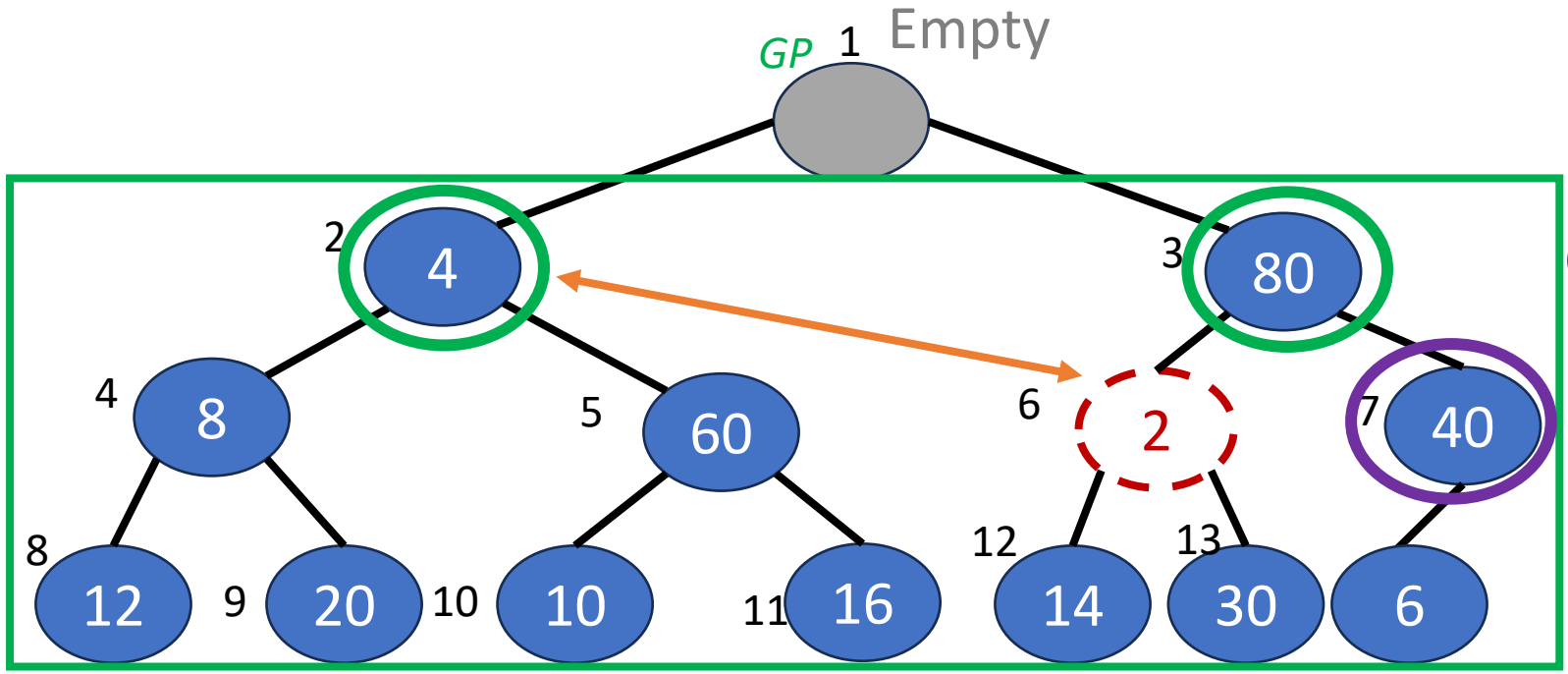
Step 2: Verify the relationship to its sibling. If violated, exchange.

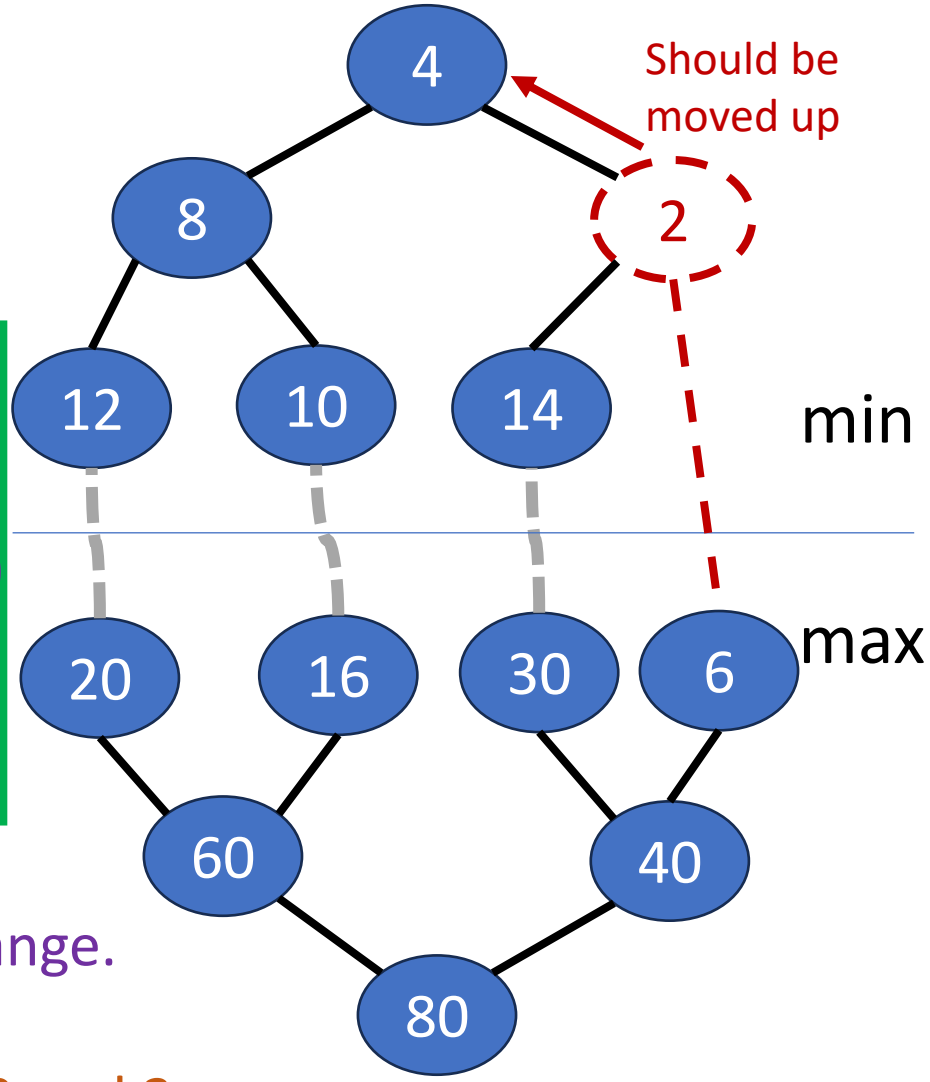
Step 3: Verify its relationship to *GP*'s children.

- Violated: exchange with that *GP*'s child and repeat steps 2 and 3.
- Not violated: terminate.

Operation: Insertion

- Example: Insert 2





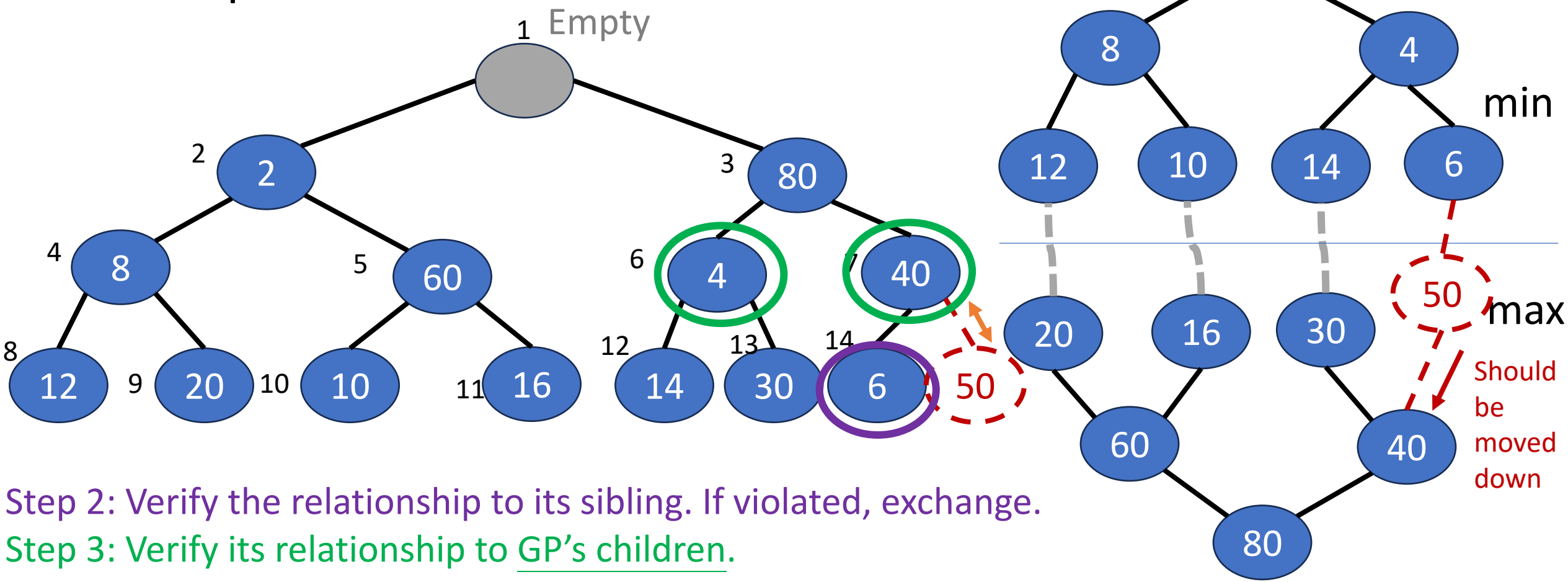
Step 2: Verify the relationship to its sibling. If violated, exchange.

Step 3: Verify its relationship to GP's children.

- Violated: exchange with that GP's child and repeat steps 2 and 3.
- Not violated: terminate.

Operation: Insertion

- Example: Insert 50



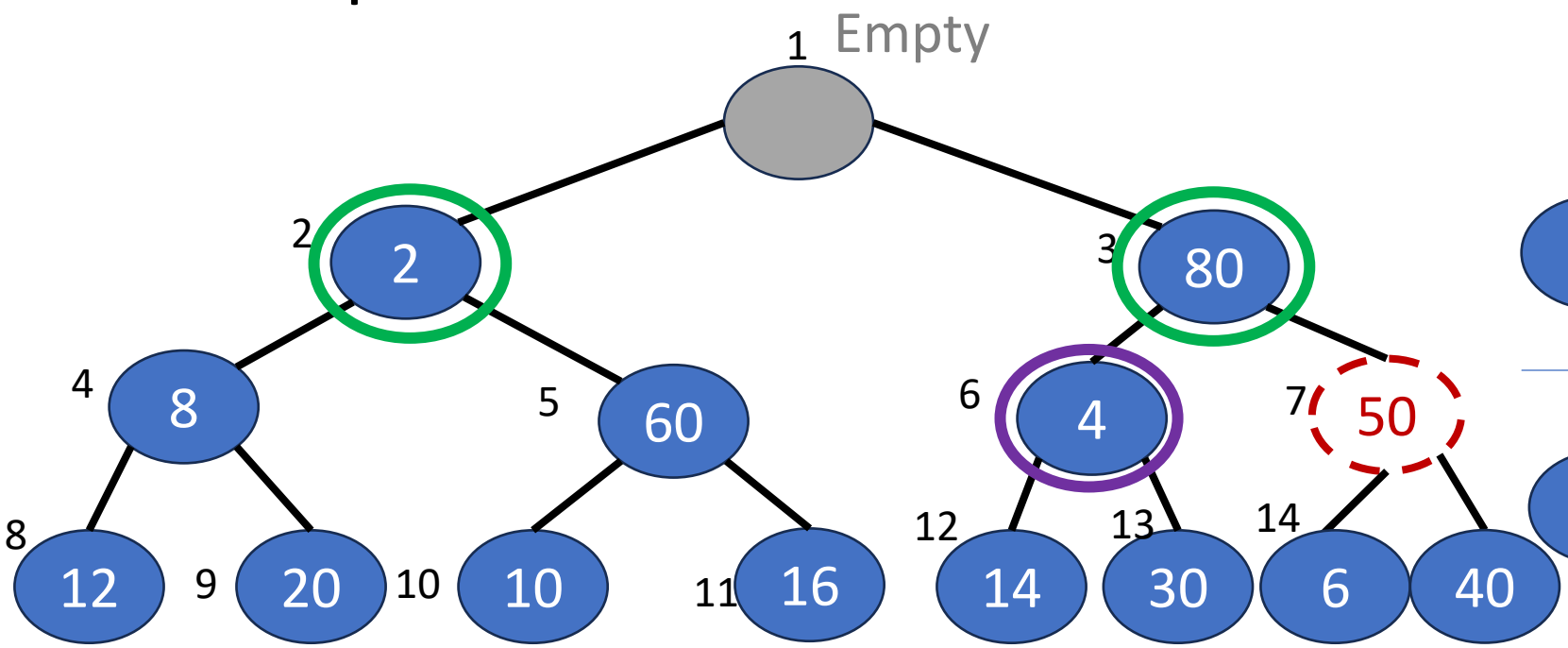
Step 2: Verify the relationship to its sibling. If violated, exchange.

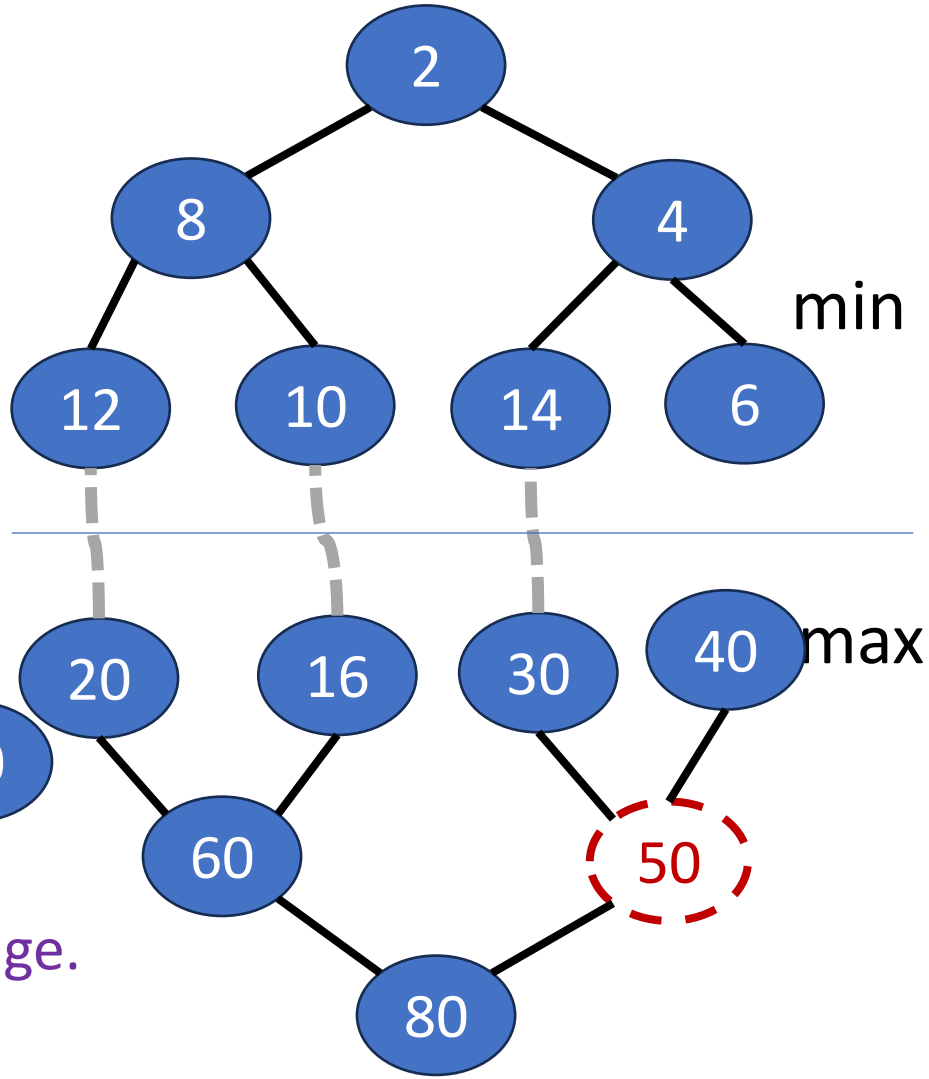
Step 3: Verify its relationship to GP's children.

- Violated: exchange with that GP's child and repeat steps 2 and 3.
- Not violated: terminate.

Operation: Insertion

- Example: Insert 50





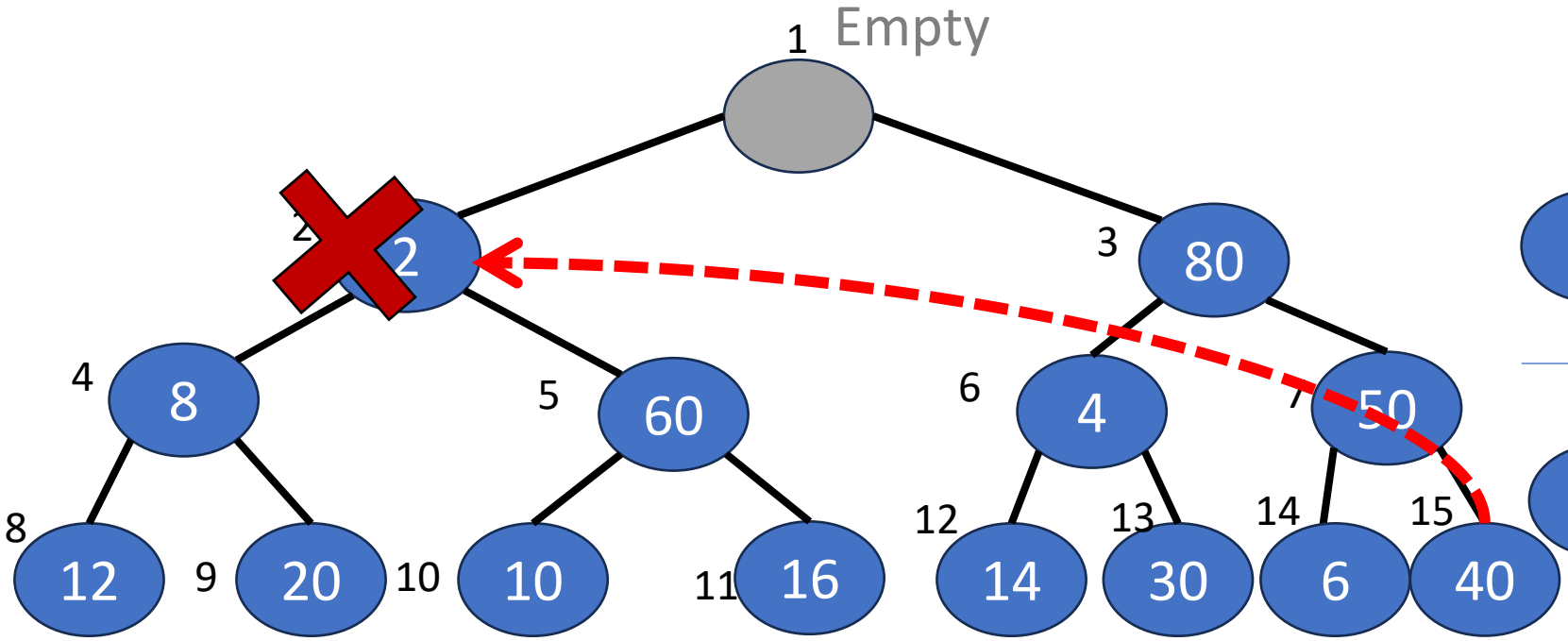
Step 2: Verify the relationship to its sibling. If violated, exchange.

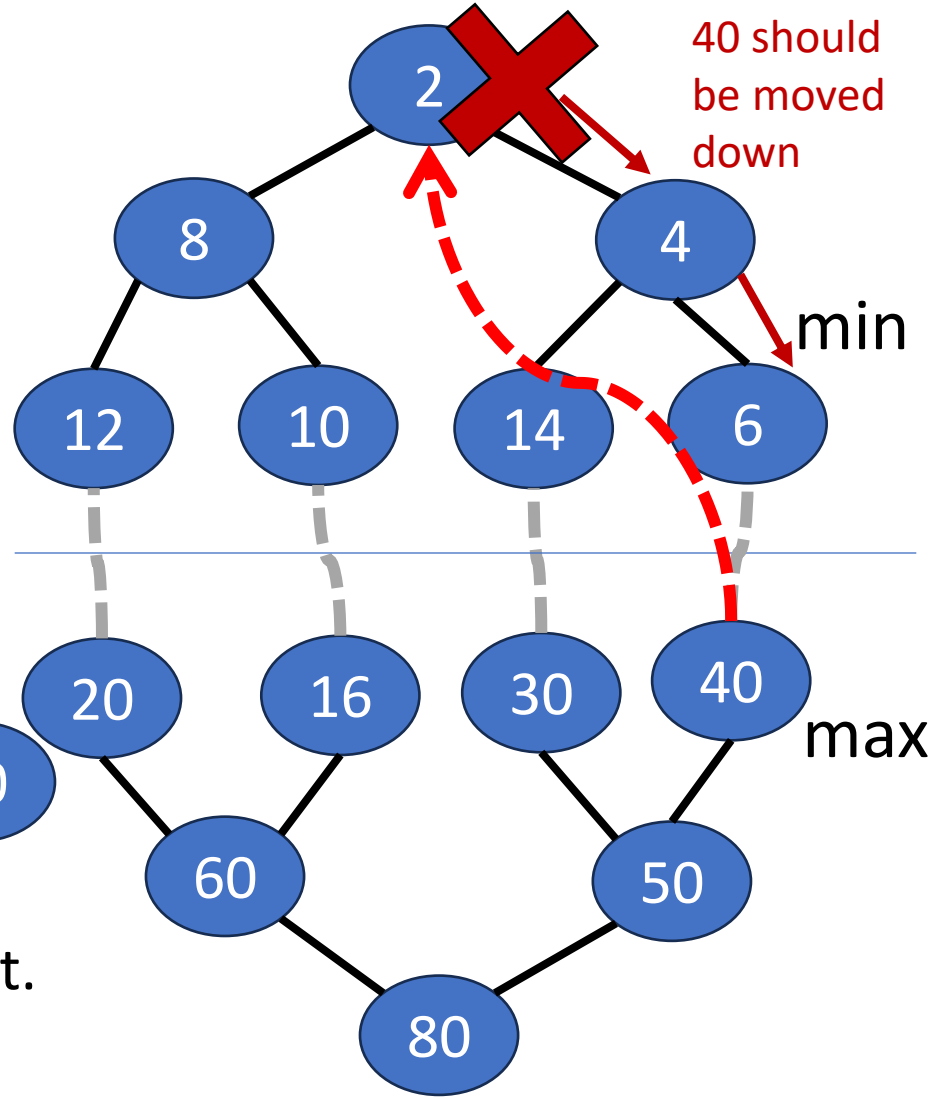
Step 3: Verify its relationship to GP's children.

- Violated: exchange with that GP's child and repeat steps 2 and 3.
- Not violated: terminate.

Operation: Deletion

- Example: Delete min element

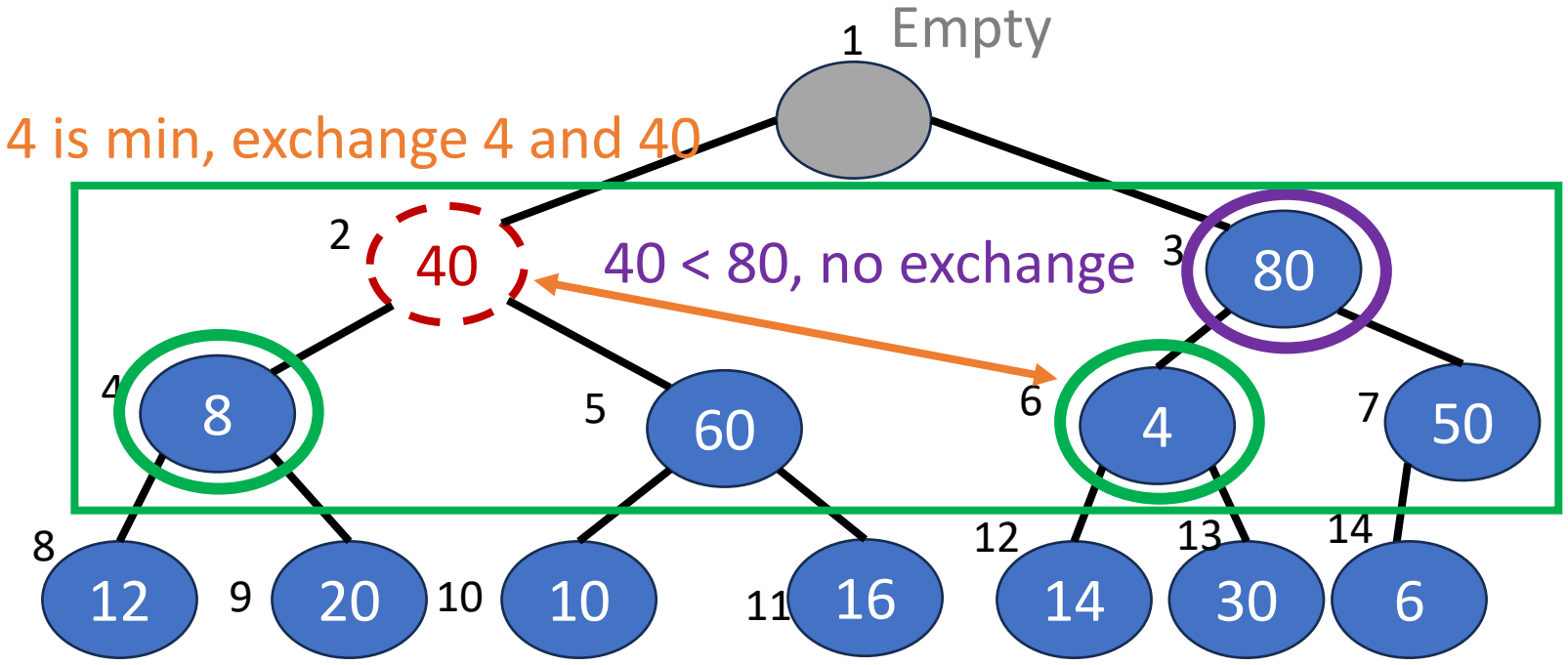


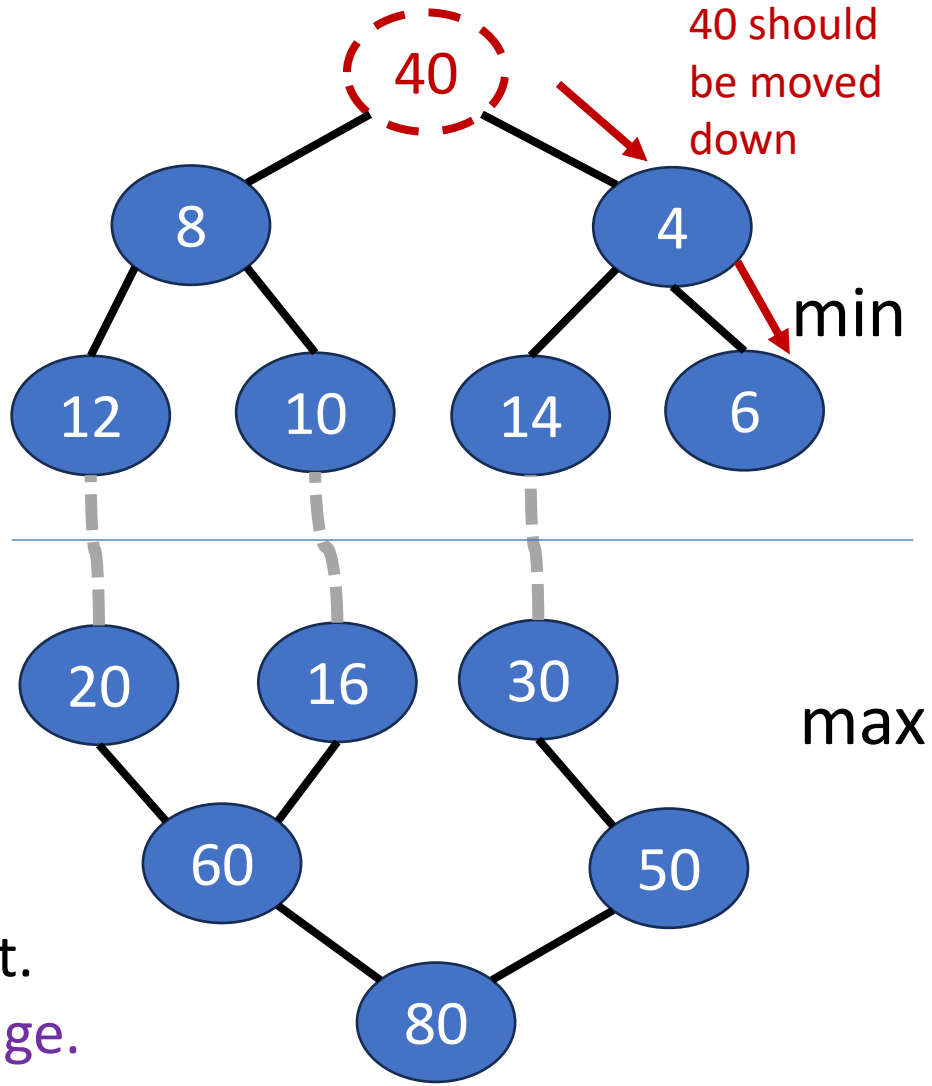


Step 1: Delete left child of the root and insert the last element.

Operation: Deletion

- Example: Delete min element





Step 1: Delete left child of the root and insert the last element.

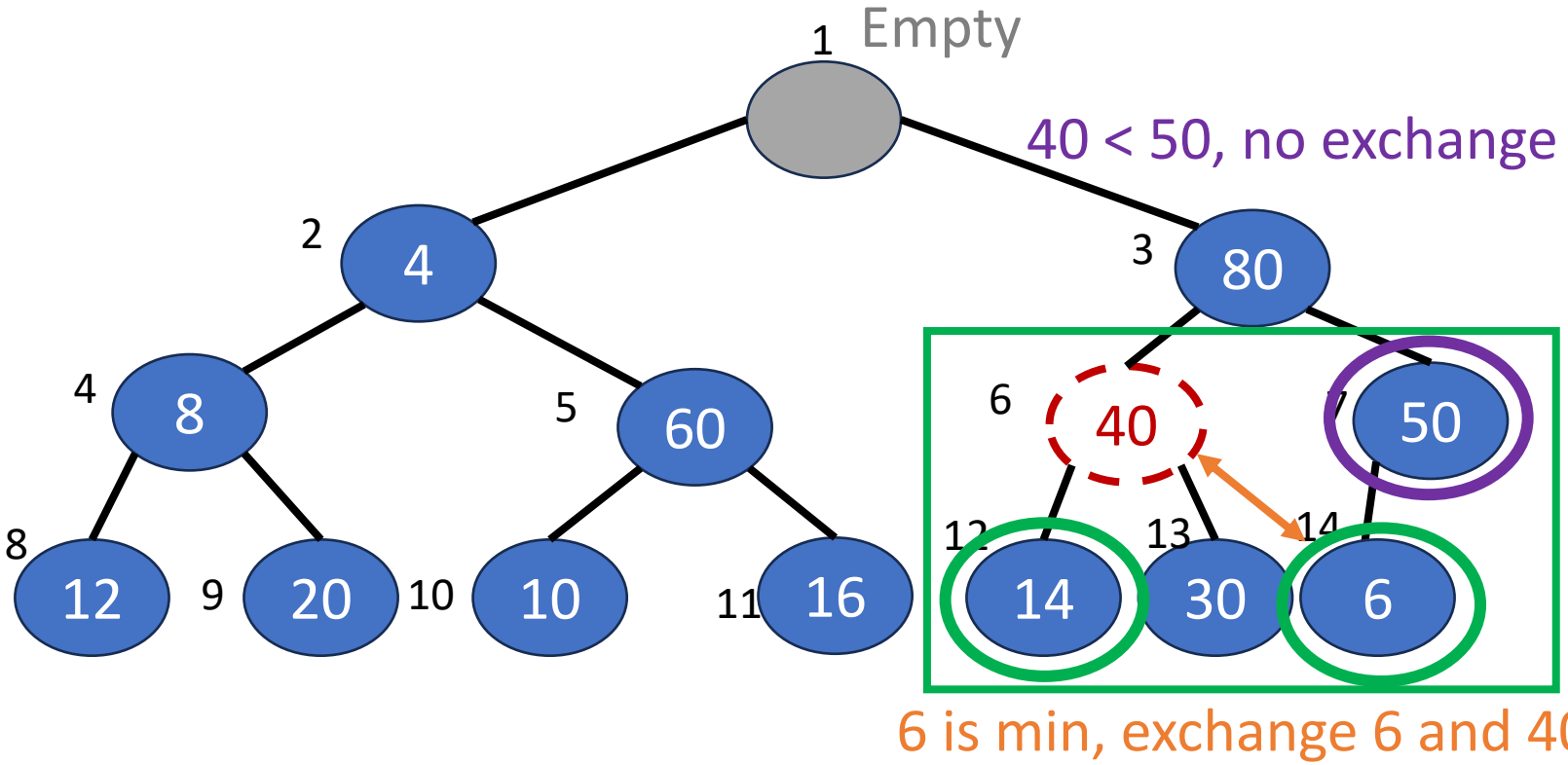
Step 2: Verify the relationship to its sibling. If violated, exchange.

Step 3: Verify its relationship to its children and its sibling's children.

- Violated: exchange with the child causes violation. Repeat steps 2 and 3.
- Not violated: terminate.

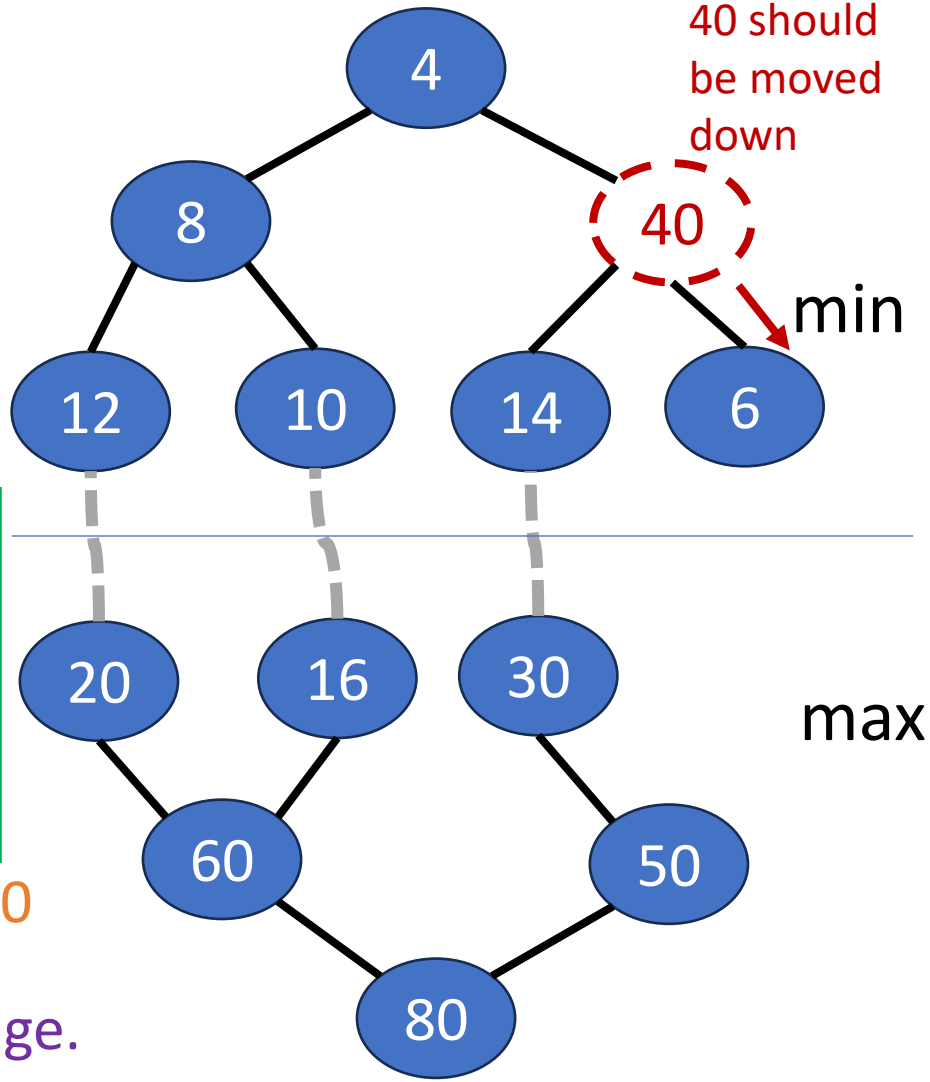
Operation: Deletion

- Example: Delete min element



40 < 50, no exchange

6 is min, exchange 6 and 40



40 should be moved down

min

max

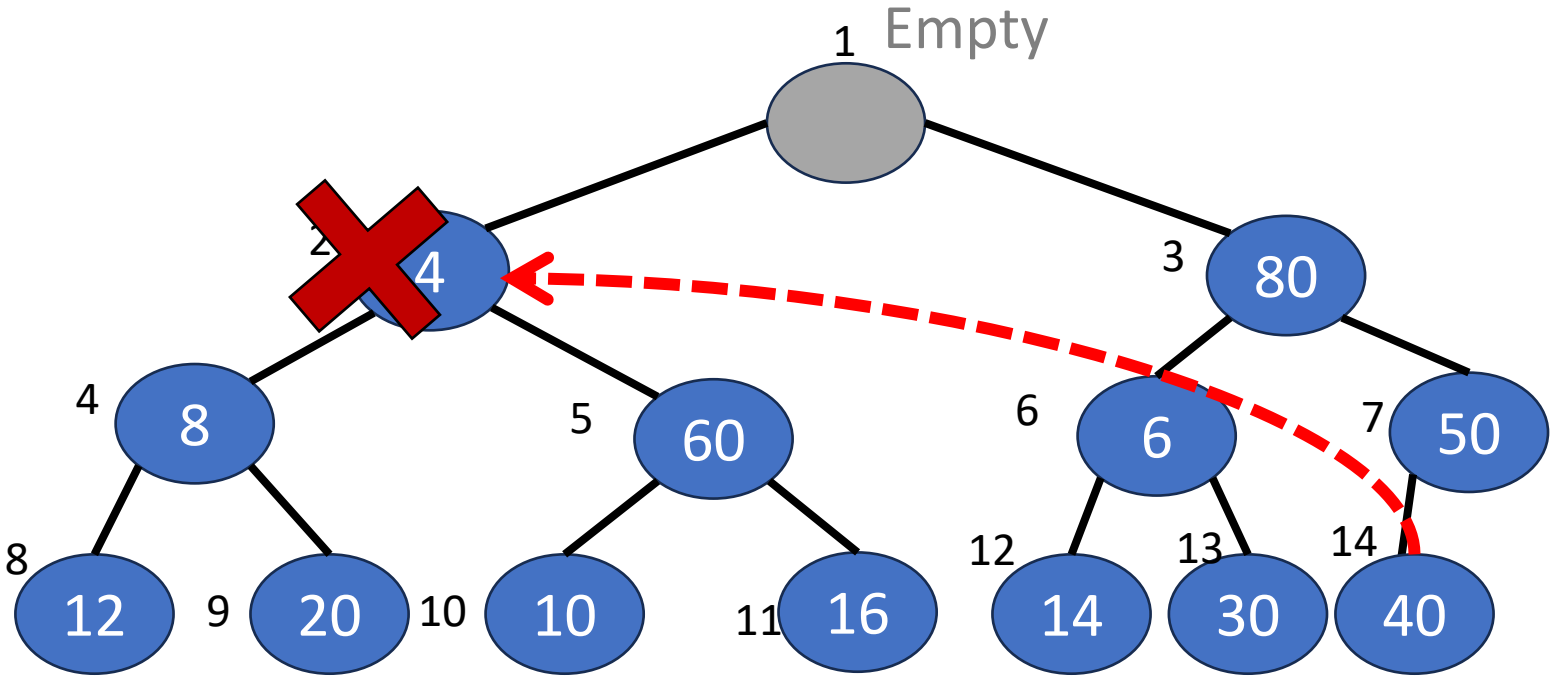
Step 2: Verify the relationship to its sibling. If violated, exchange.

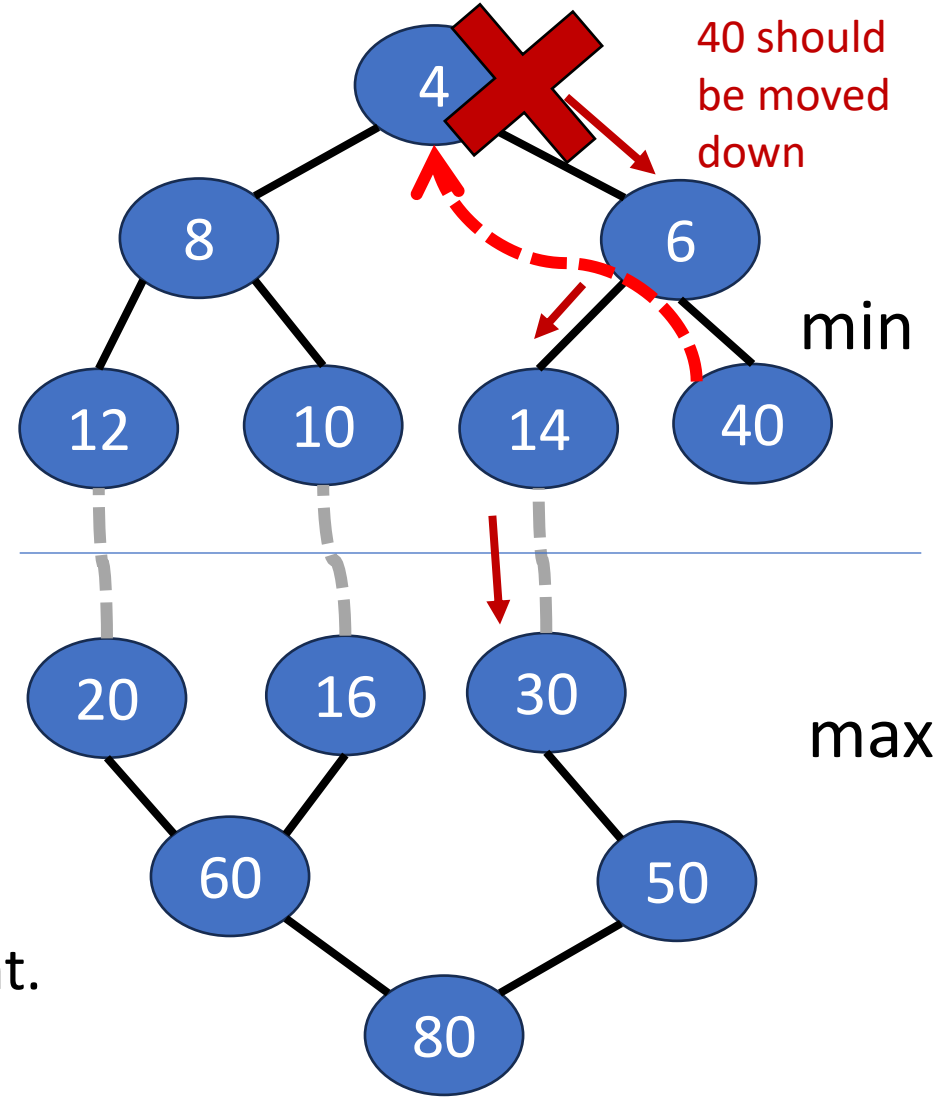
Step 3: Verify its relationship to its children and its sibling's children.

- Violated: exchange with the child causes violation. Repeat steps 2 and 3.
- Not violated: terminate.

Operation: Deletion

- Example: Delete min element

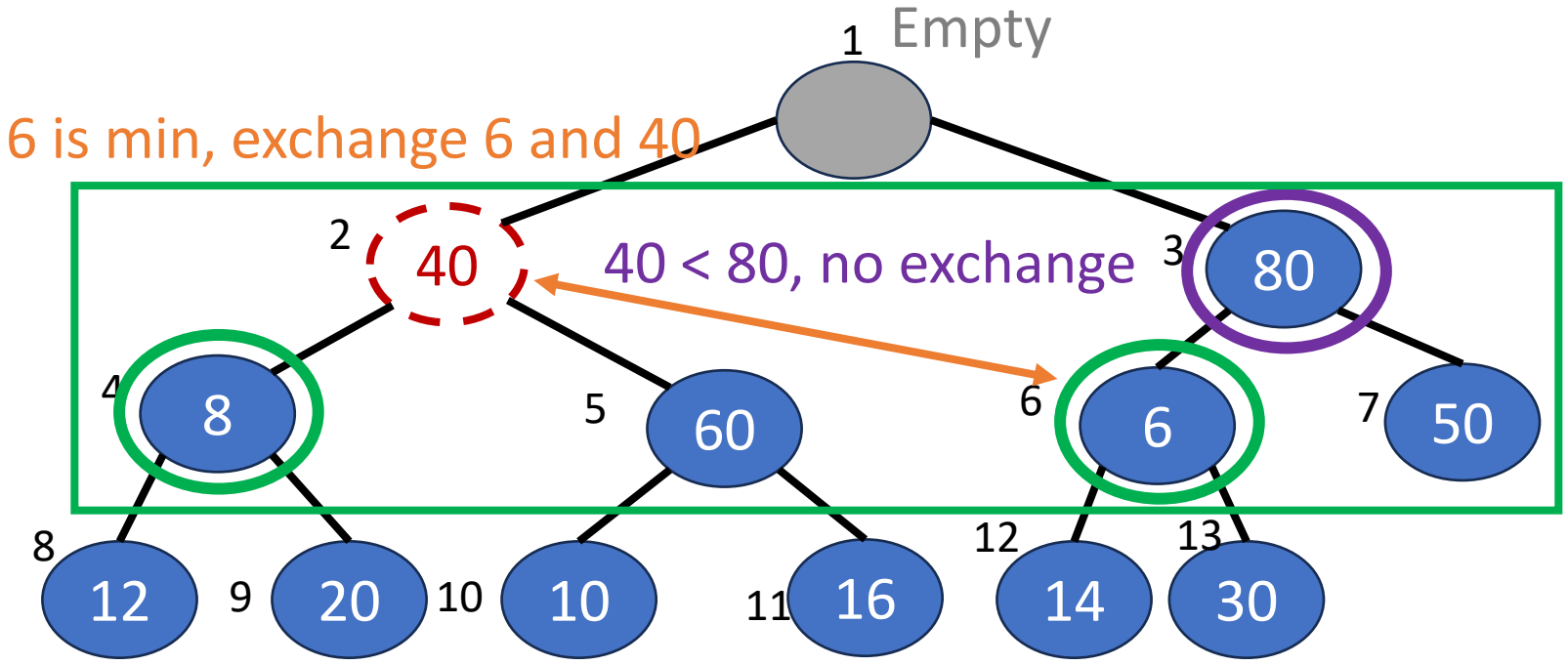


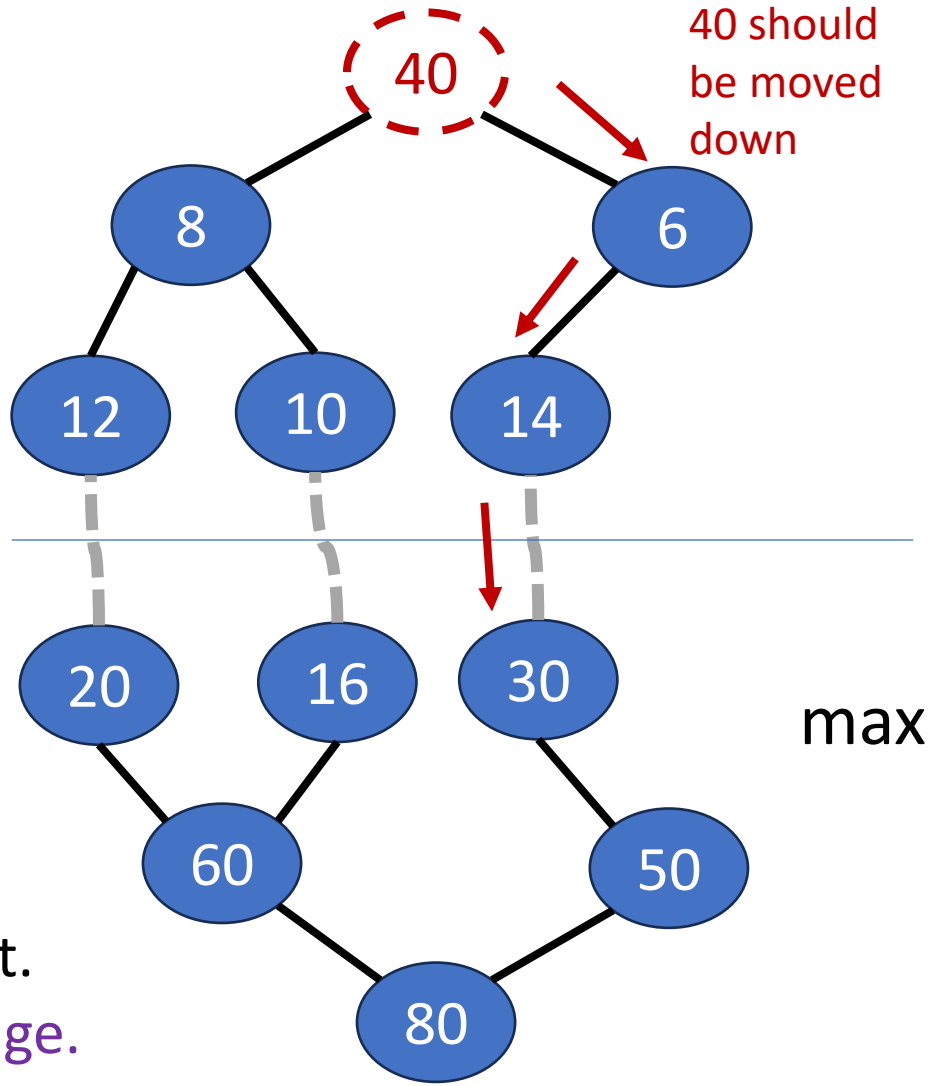


Step 1: Delete left child of the root and insert the last element.

Operation: Deletion

- Example: Delete min element





Step 1: Delete left child of the root and insert the last element.

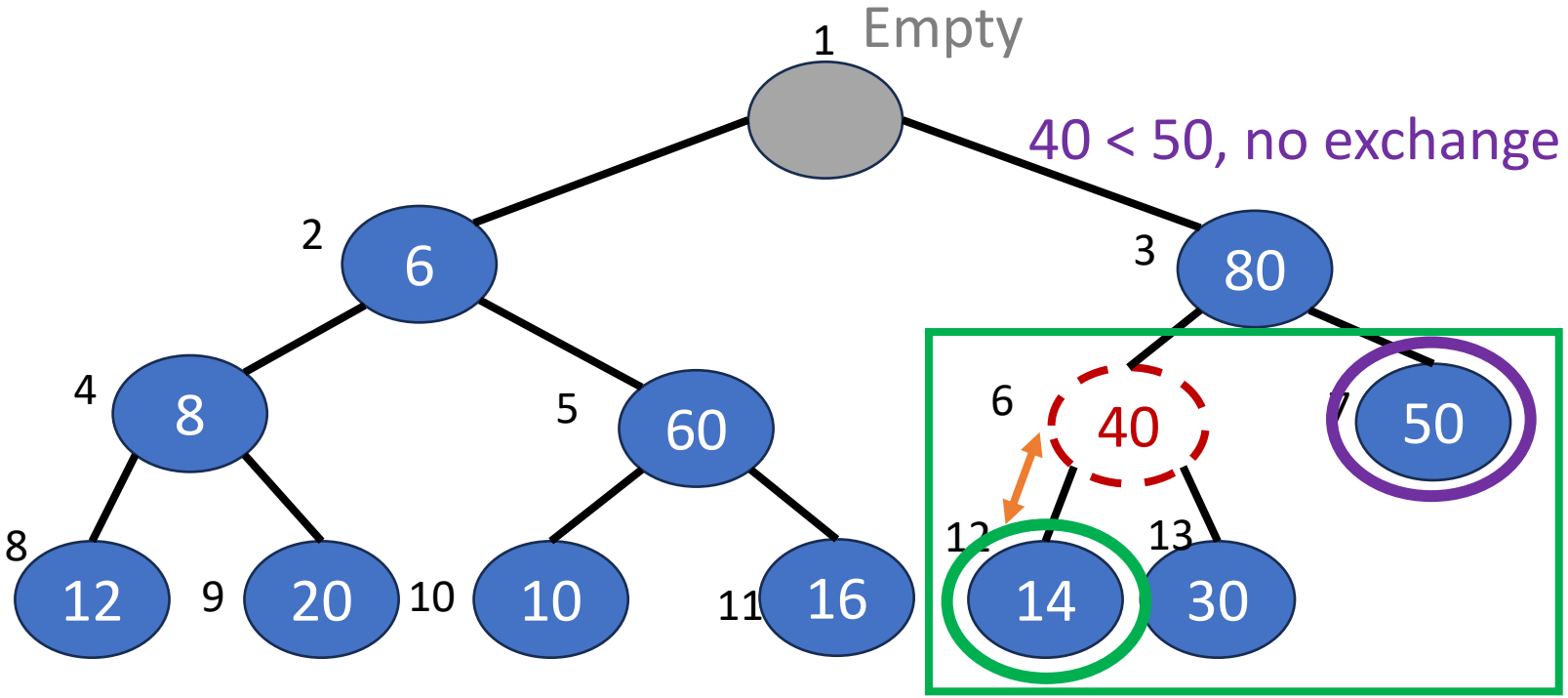
Step 2: Verify the relationship to its sibling. If violated, exchange.

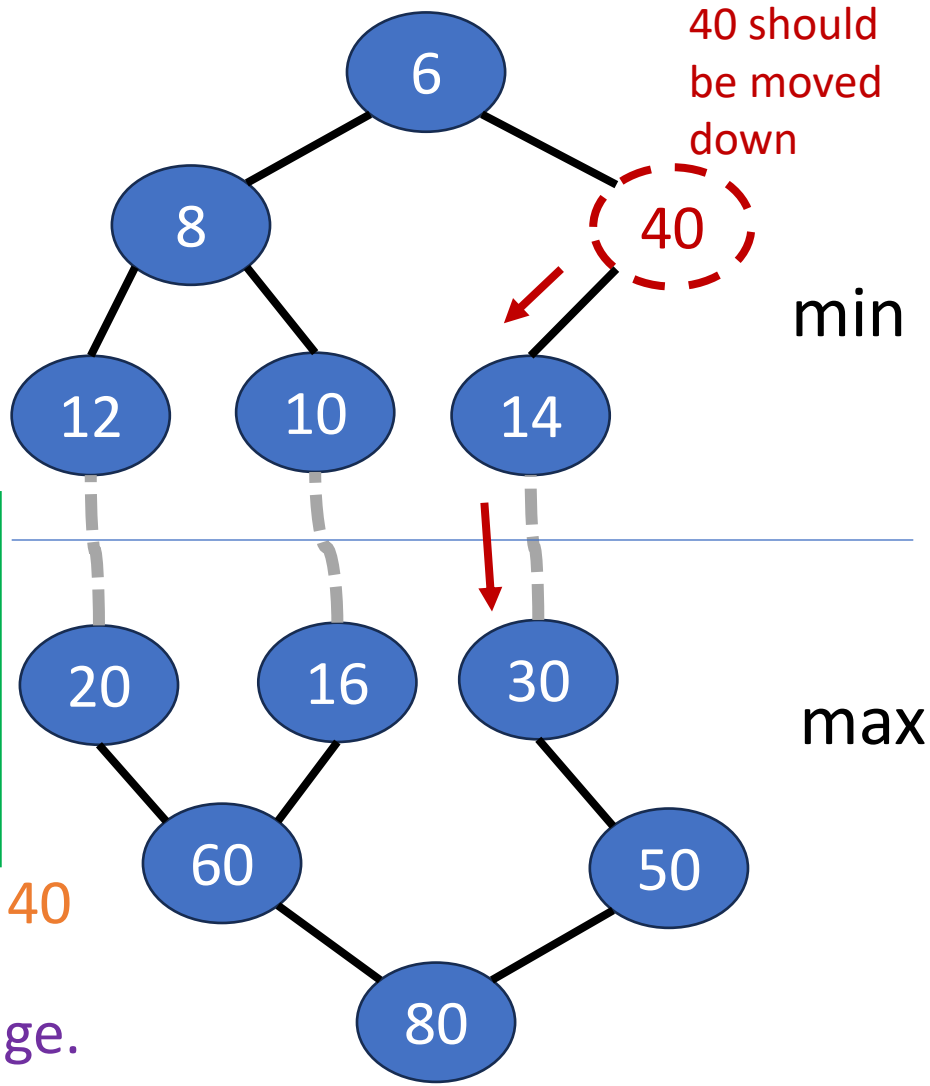
Step 3: Verify its relationship to its children and its sibling's children.

- Violated: exchange with the child causes violation. Repeat steps 2 and 3.
- Not violated: terminate.

Operation: Deletion

- Example: Delete min element





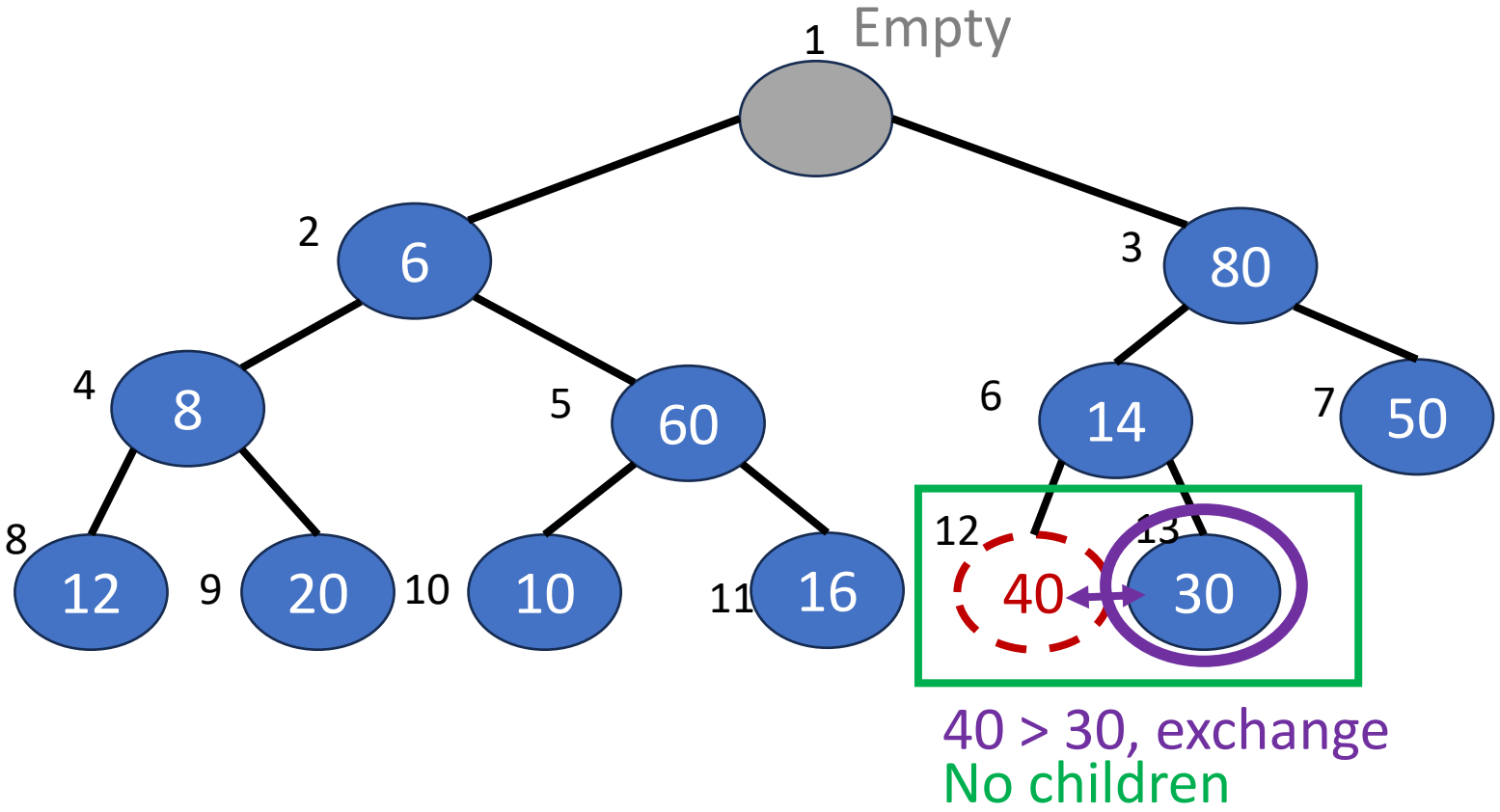
Step 2: Verify the relationship to its sibling. If violated, exchange.

Step 3: Verify its relationship to its children and its sibling's children.

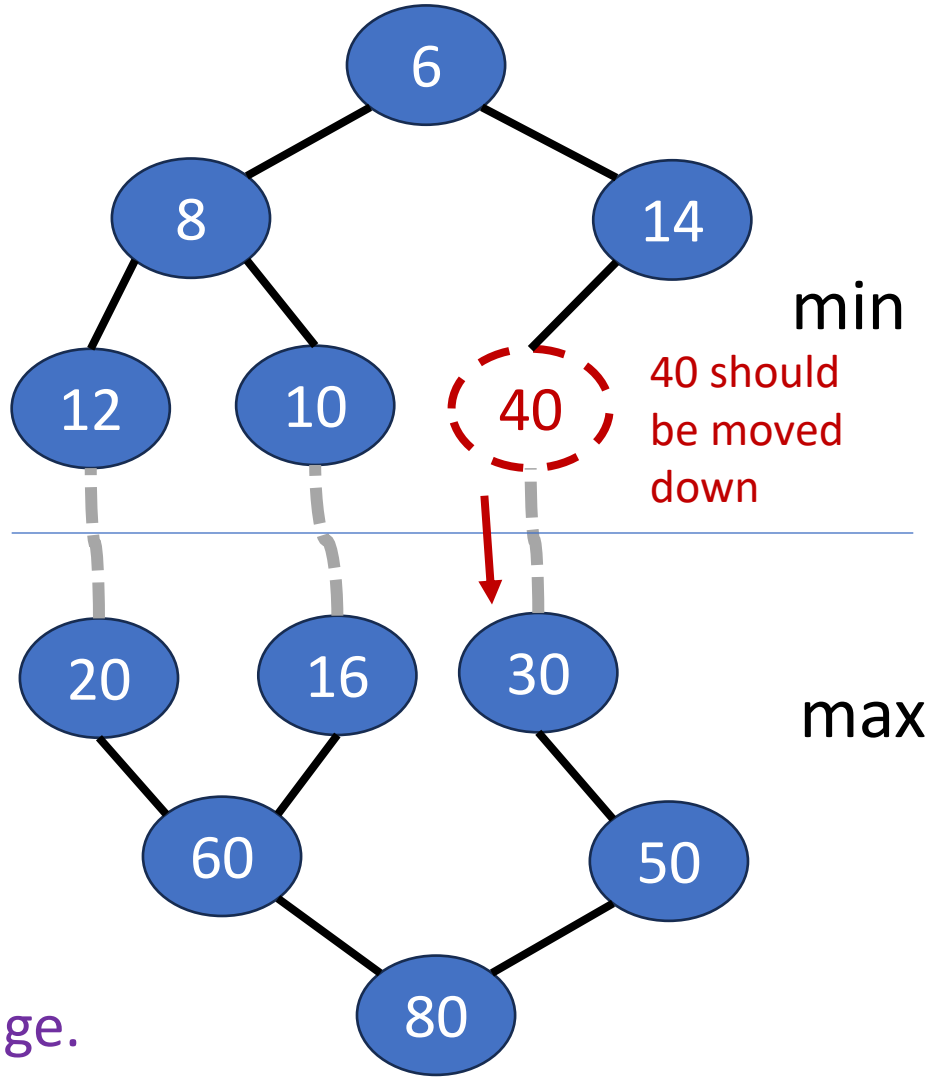
- Violated: exchange with the child causes violation. Repeat steps 2 and 3.
- Not violated: terminate.

Operation: Deletion

- Example: Delete min element



40 > 30, exchange
No children



Step 2: Verify the relationship to its sibling. If violated, exchange.

Step 3: Verify its relationship to its children and its sibling's children.

- Violated: exchange with the child causes violation. Repeat steps 2 and 3.
- Not violated: terminate.

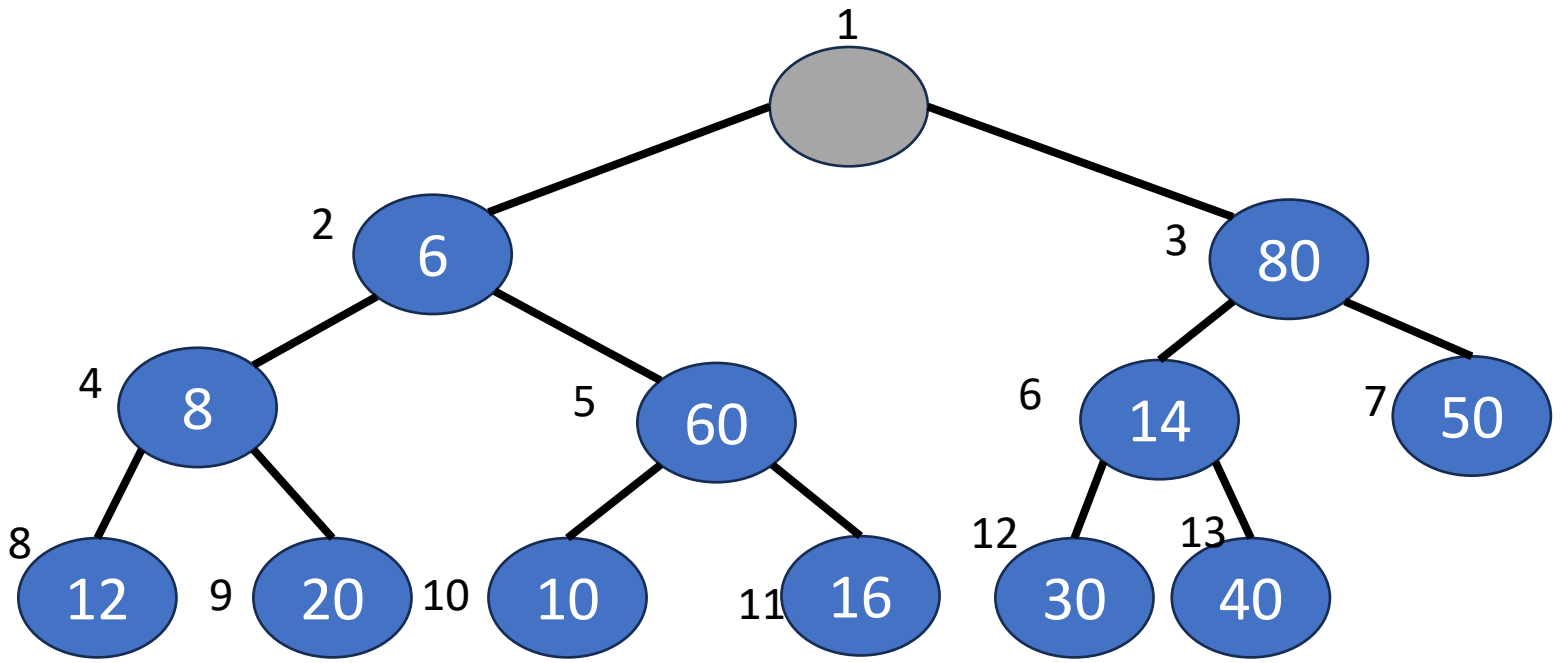
Time complexity

- Insertion:
  - O(1) time at each level during the bubble-up pass
  - Total: O(log n)

- Delete min:
  - O(1) time at each level during the trickle-down pass
  - Total: O(log n)

Exercise

- Q4: Perform 3 delete-min operation on the following SMMH. Where will be the location of 60?
- Q5: Perform a delete-max operation on the following SMMH. Where will be the location of 30?



Please reply your answers of Q4-5 via the following link:

https://forms.gle/aWUwuR7JjTtyMCMv7

Group members: 2~4 people

Exercise

- Given the elements 20, 10, 40, 3, 2, 7, 60, 1, and 80 (in this order).
  - Q6: Insert all elements sequentially into an empty min-max heap.
  - Q7: Insert all elements sequentially into an empty deap.
  - Q8: Insert all elements sequentially into an empty SMMH.

Note: Please write out the resultant heap using array representation (start at index 0).

Please reply your answers of Q6-8 via the following link:

https://forms.gle/aWUwuR7JjTtyMCMv7

Group members: 2~4 people

Summary

- Three types of DEPQ:
  - Min-max heap
  - Doubly ended heap (Deap)
  - Symmetric min-max heap (SMMH)

- Operations:
  - Get min
  - Get max
  - Insert
  - Delete min
  - Delete max