Heapsort

Koffman & Wolfgang

- kapitel 8, avsnitt 8

Heapsort

Mergesort har komplexitet O(n log n)

- men det kräver temporärt ett extra fält

Heapsort kräver inte något extraminne

- det är en in-place-algoritm
- som namnet antyder använder algoritmen en heap för att sortera fältet

Första versionen av Heapsort

En enkel Heapsort-algoritm:

- skapa en tom heap
- tag ett element i taget ur fältet och stoppa in i heapen
- tag sedan ut elementen ur heapen ett i taget och stoppa tillbaka i fältet

Denhär versionen kräver extraminne

- hur gör man detta in-place?

Heapsort in-place

Antag att vi har ett fält med n element

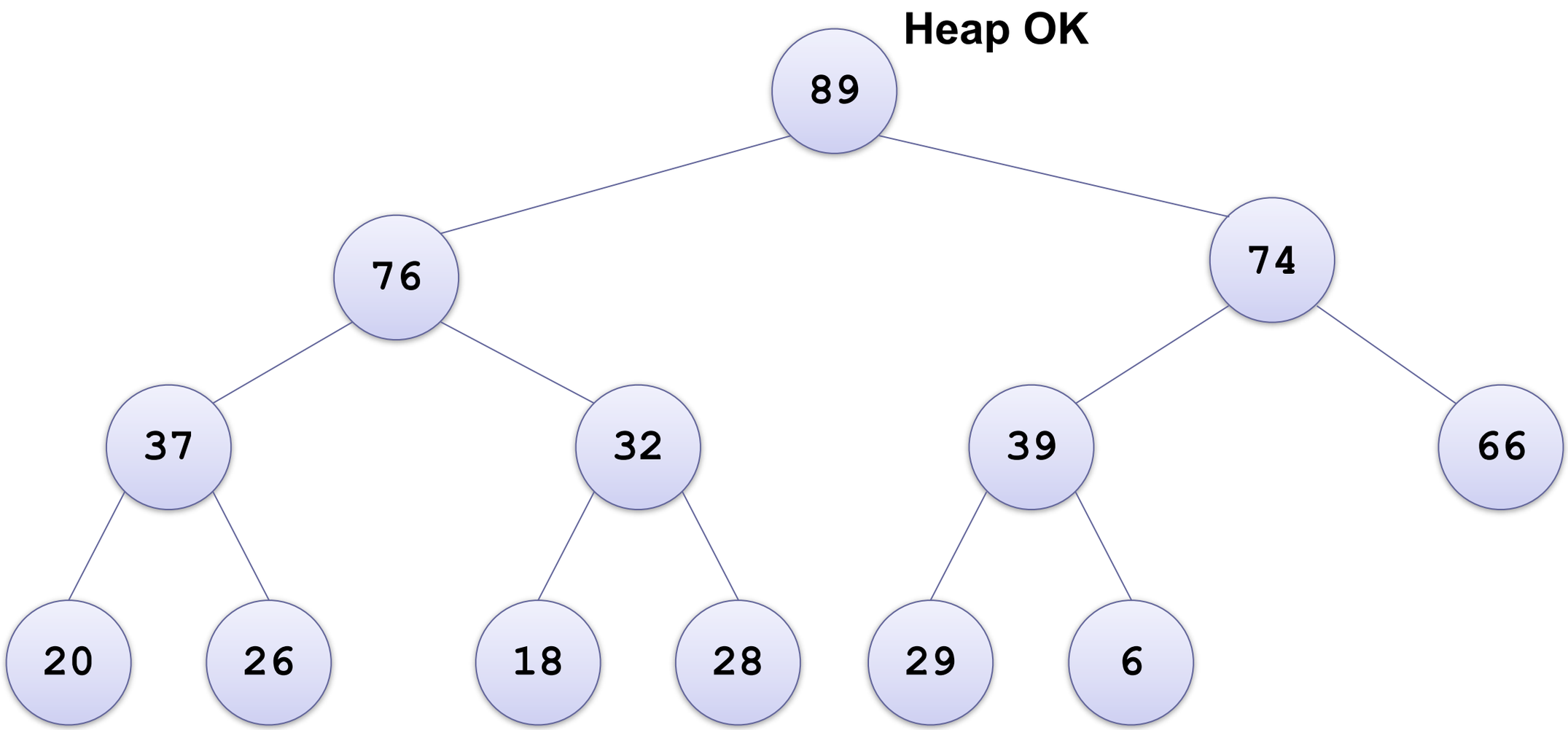
- låt första elementet vara en heap med ett element
- utöka heapen med ett element i taget
- låt elementet bubbla upp till rätt plats i heapen

Nu har vi en heap; hur gör vi den till en lista?

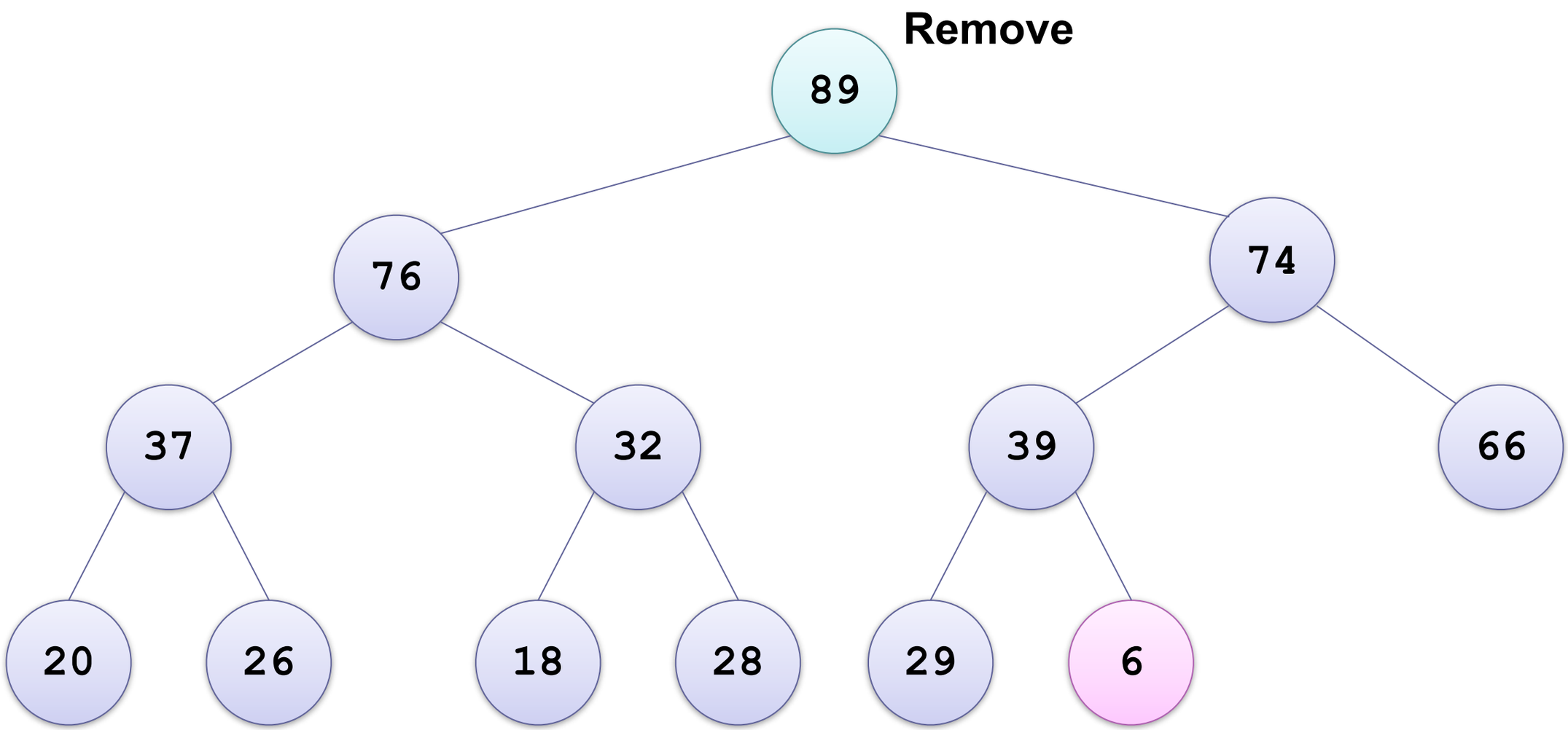
- byt ut det översta elementet med det sista
- minska heapens storlek, och låt det nya översta elementet bubbla ner till rätt ställe
- fortsätt så tills heapen bara har ett element

Vi måste bygga heapen som en max-heap,
eftersom vi flyttar elementen sist i fältet hela tiden

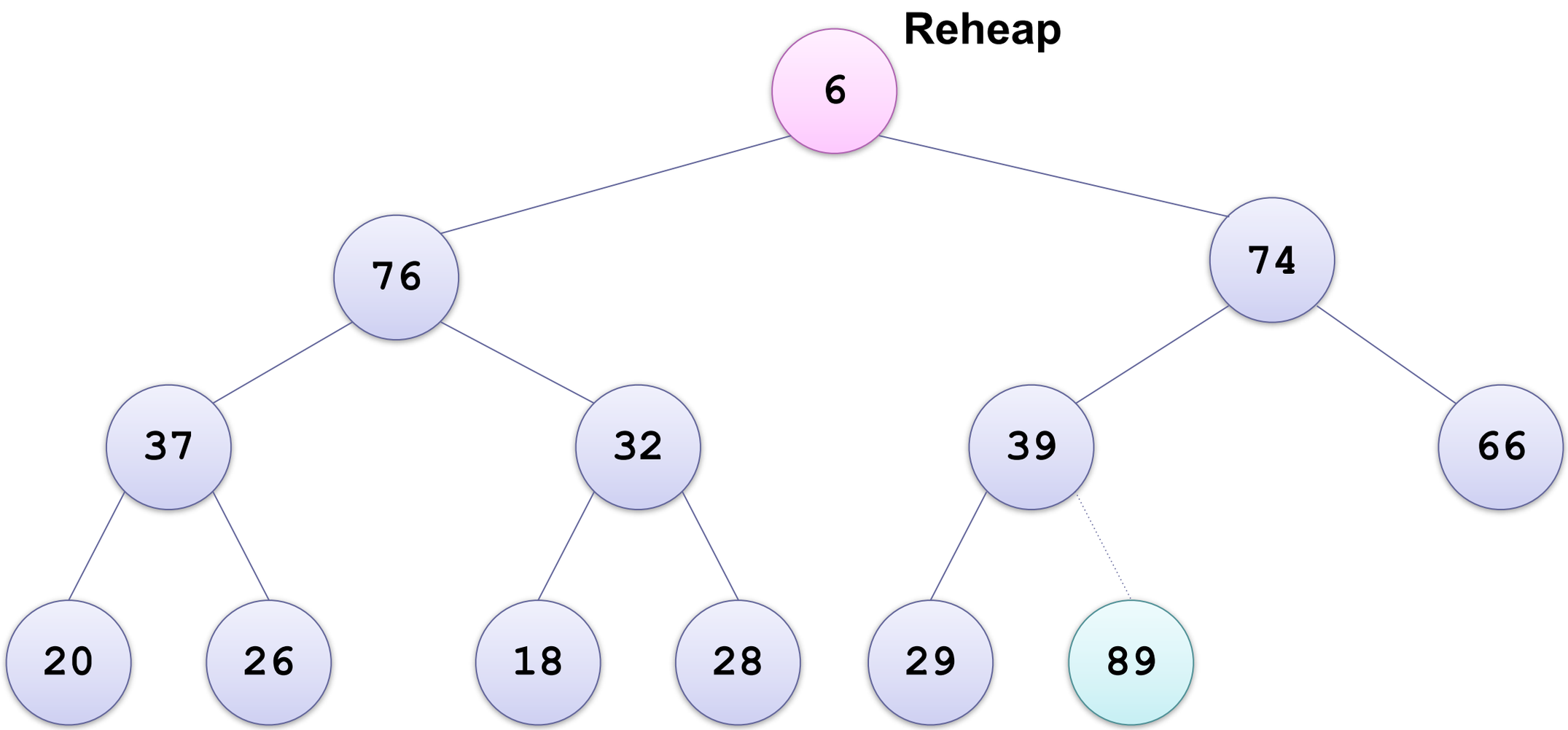
Trace of Heapsort



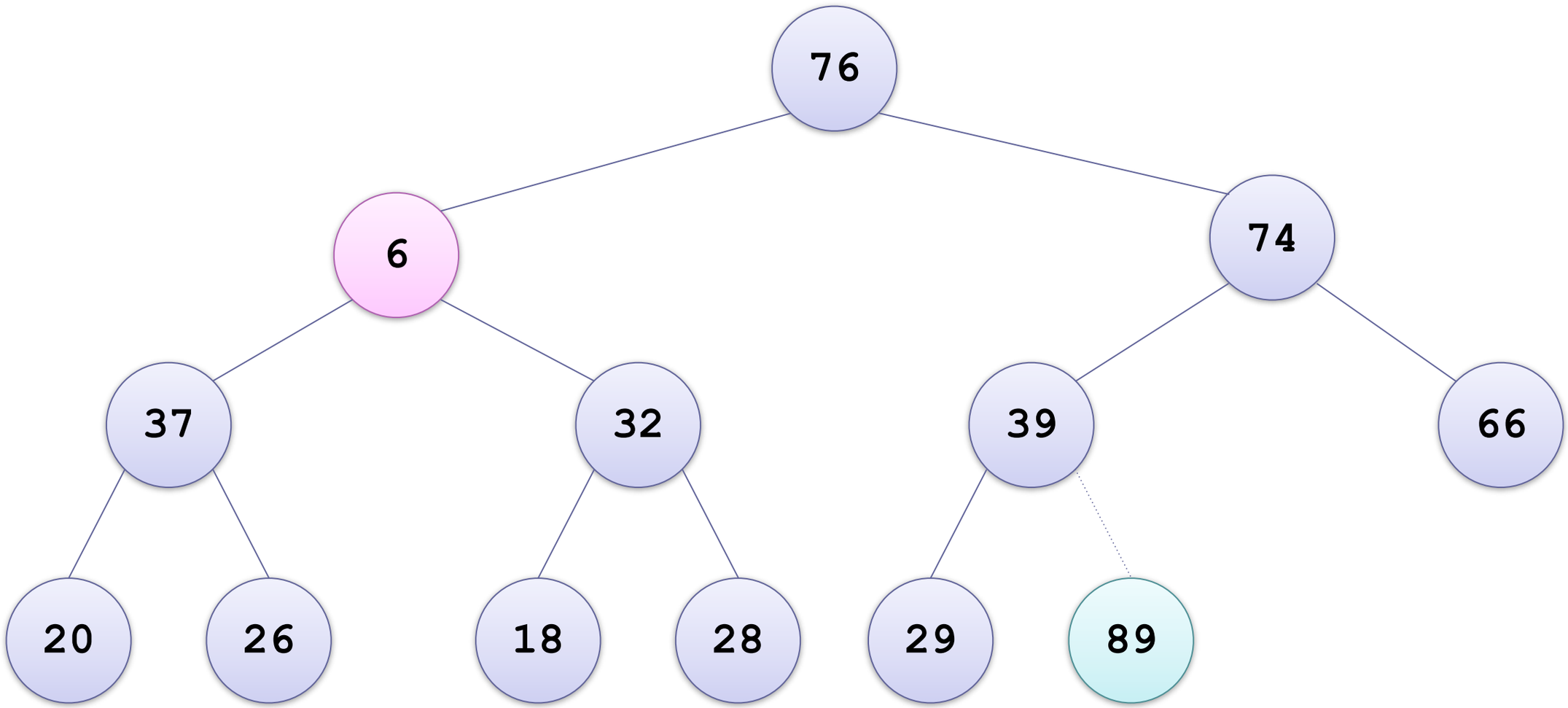
Trace of Heapsort (cont.)



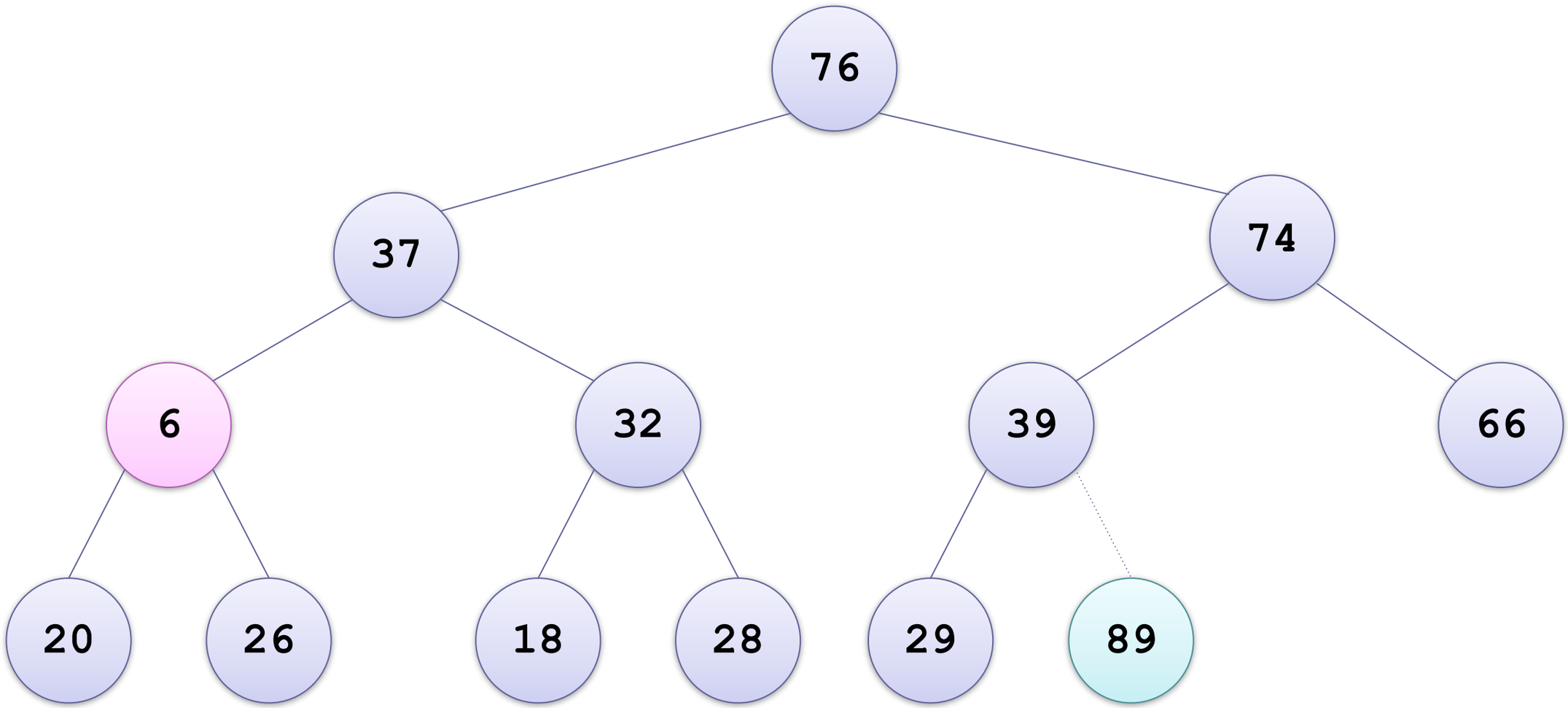
Trace of Heapsort (cont.)



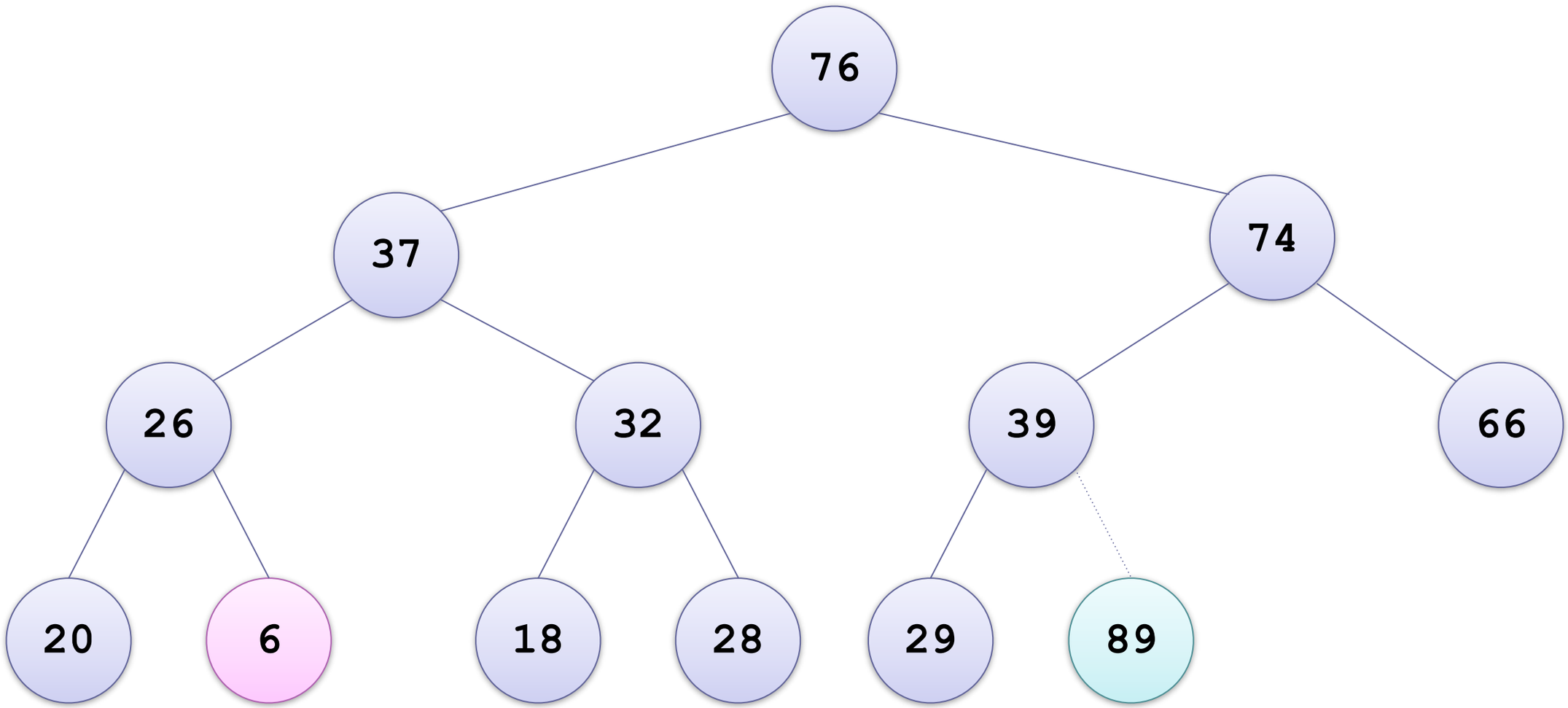
Trace of Heapsort (cont.)



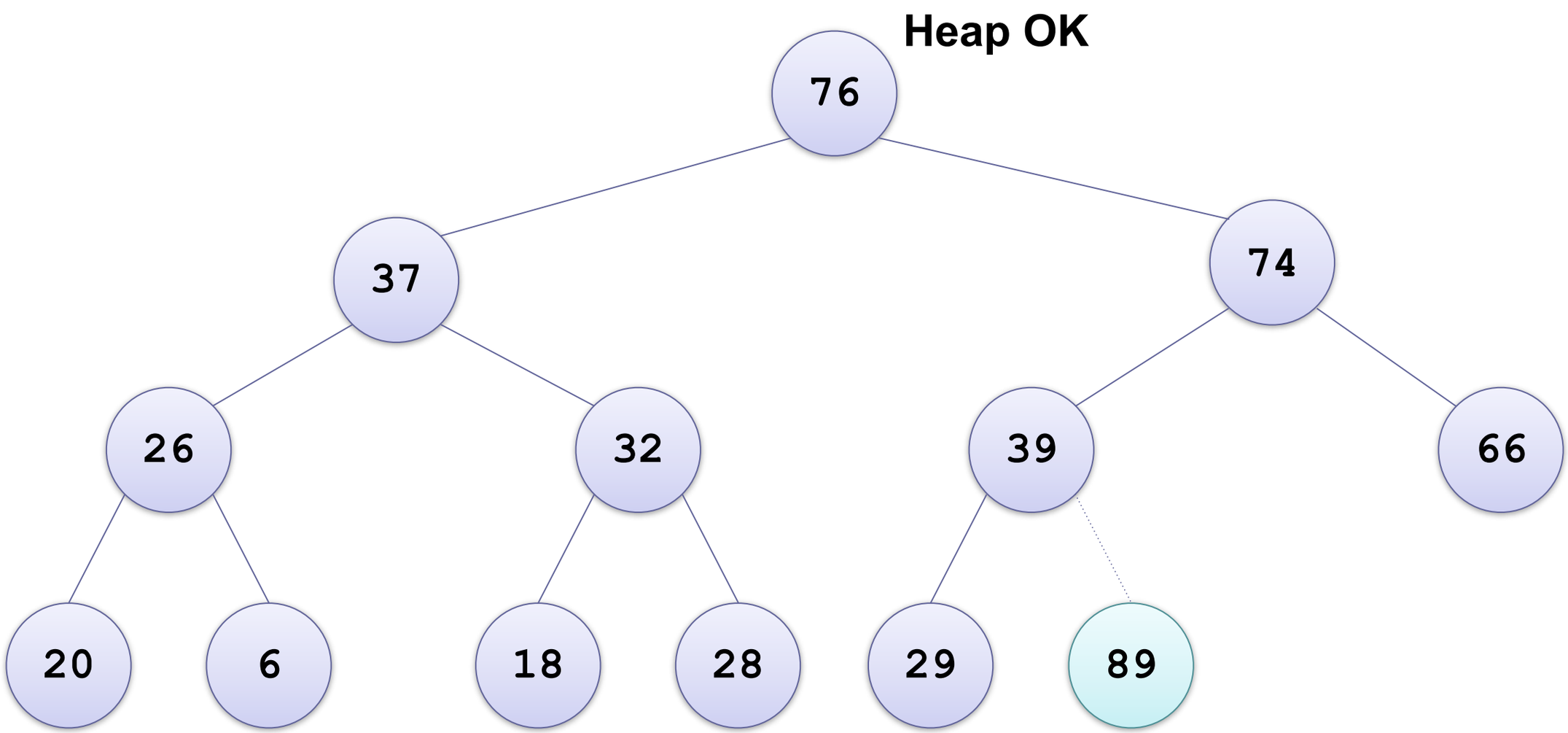
Trace of Heapsort (cont.)



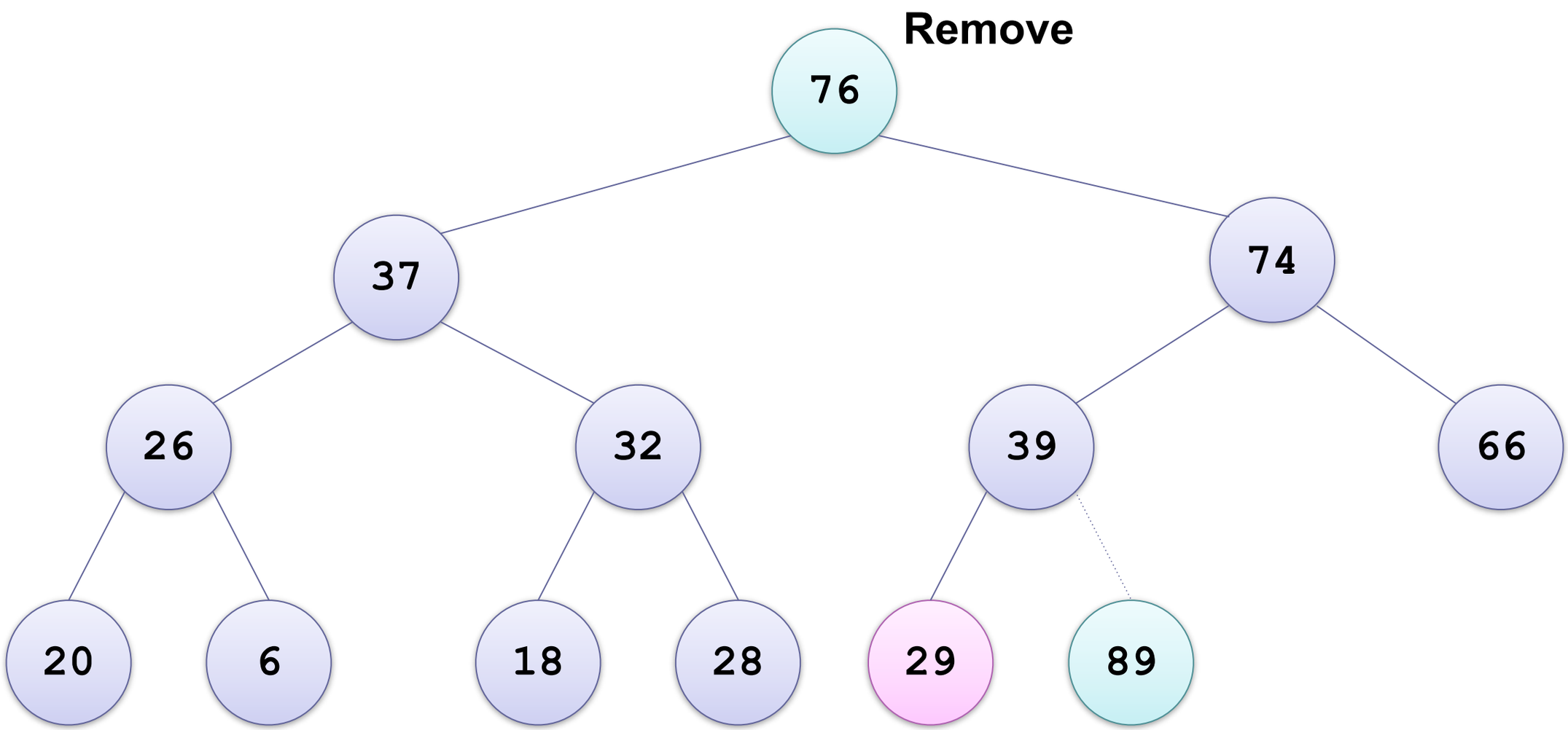
Trace of Heapsort (cont.)



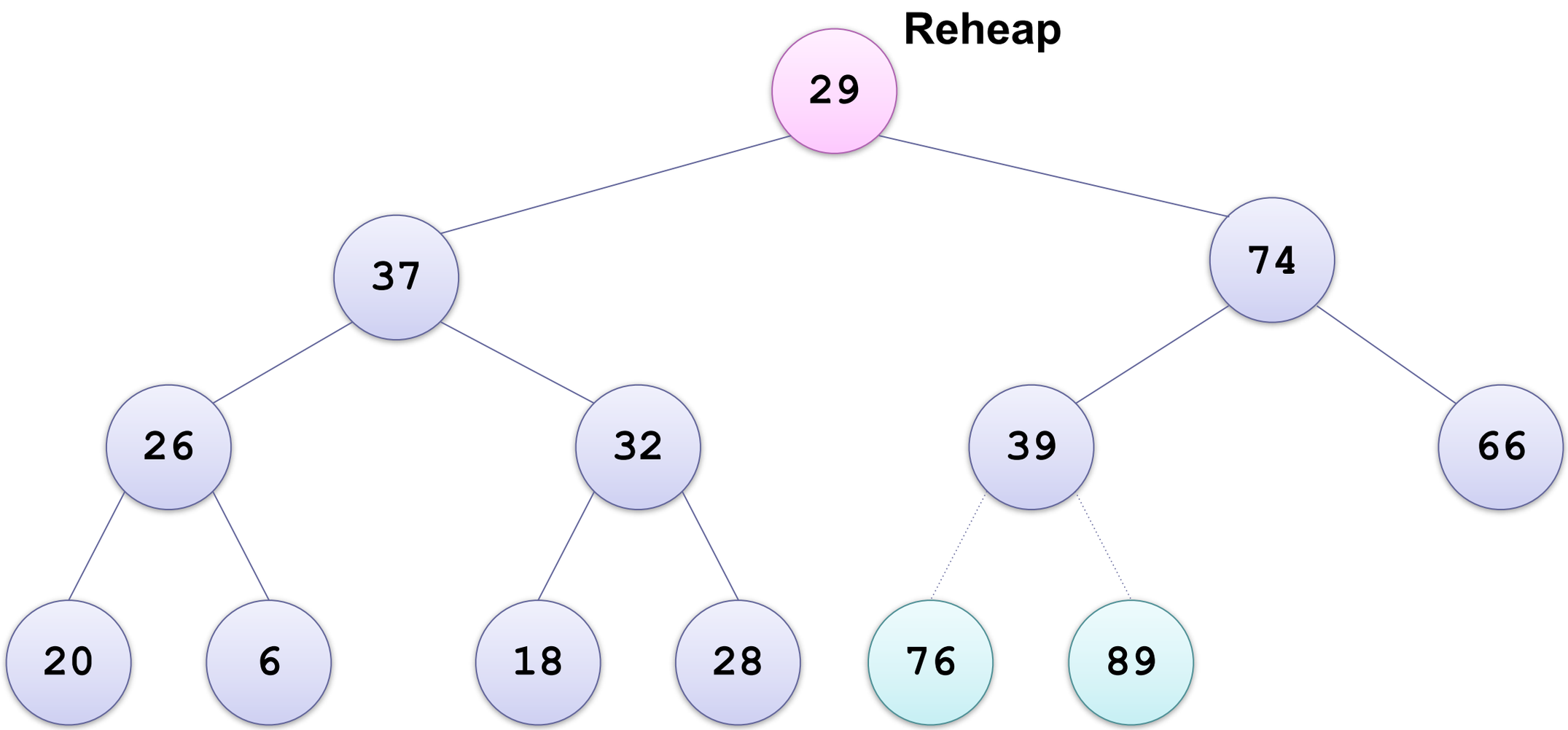
Trace of Heapsort (cont.)



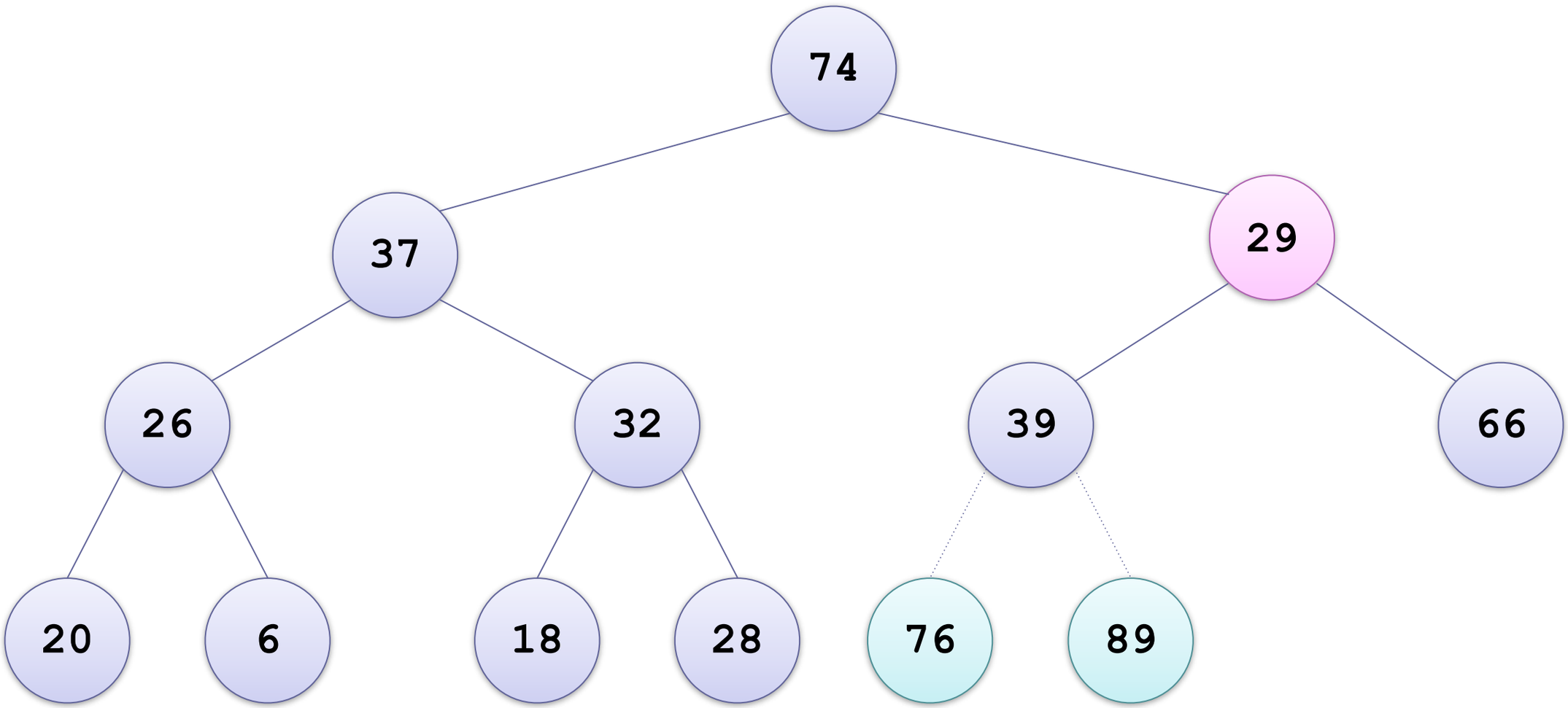
Trace of Heapsort (cont.)



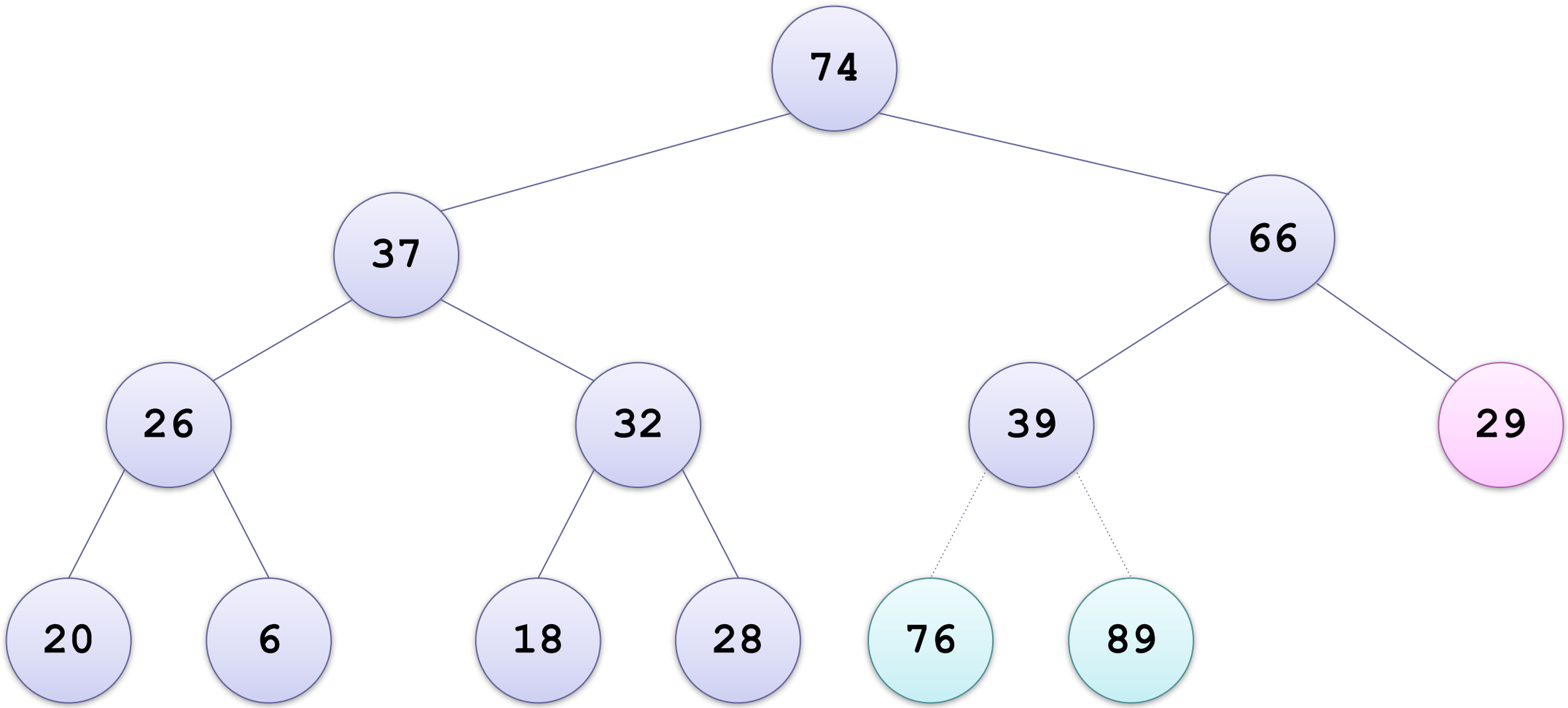
Trace of Heapsort (cont.)



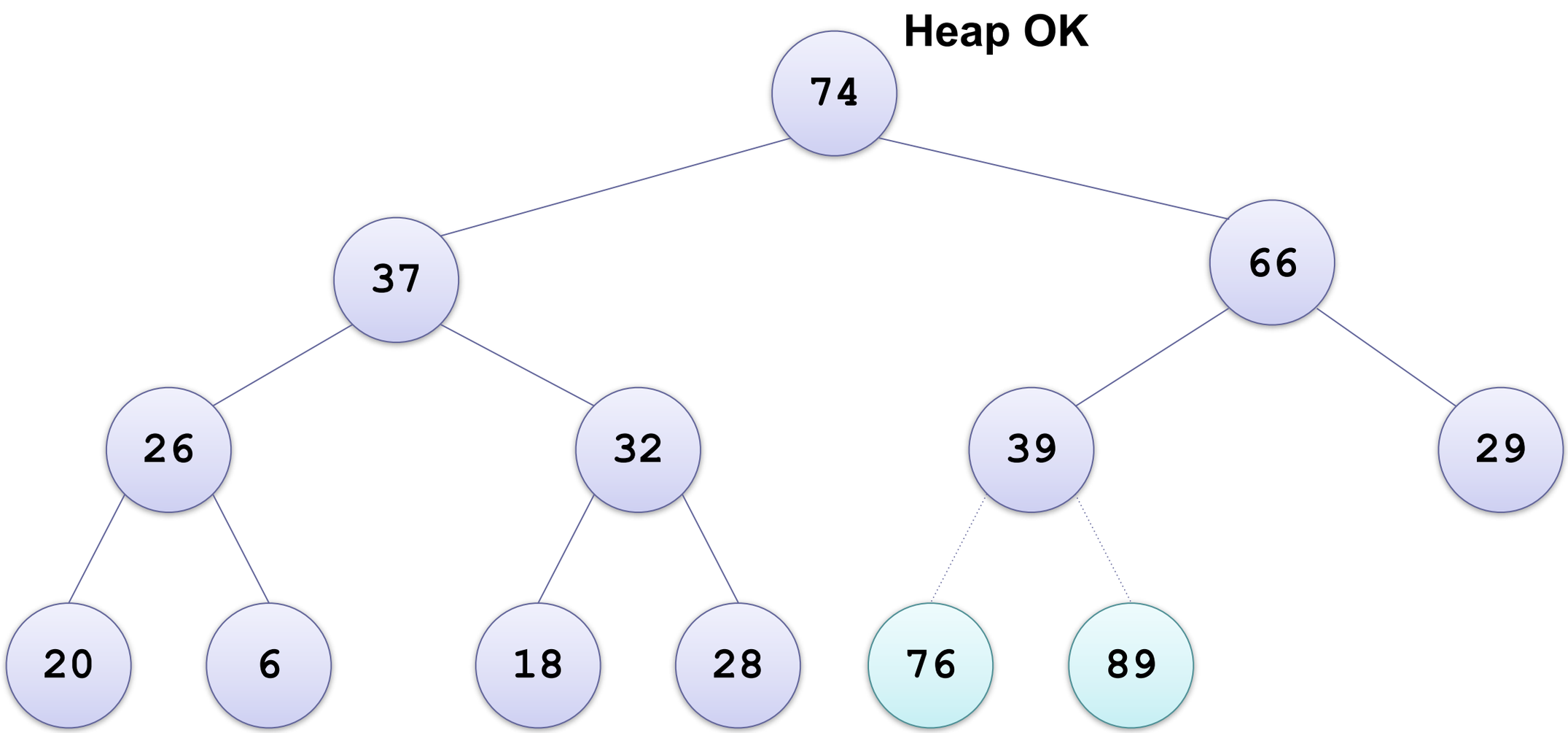
Trace of Heapsort (cont.)



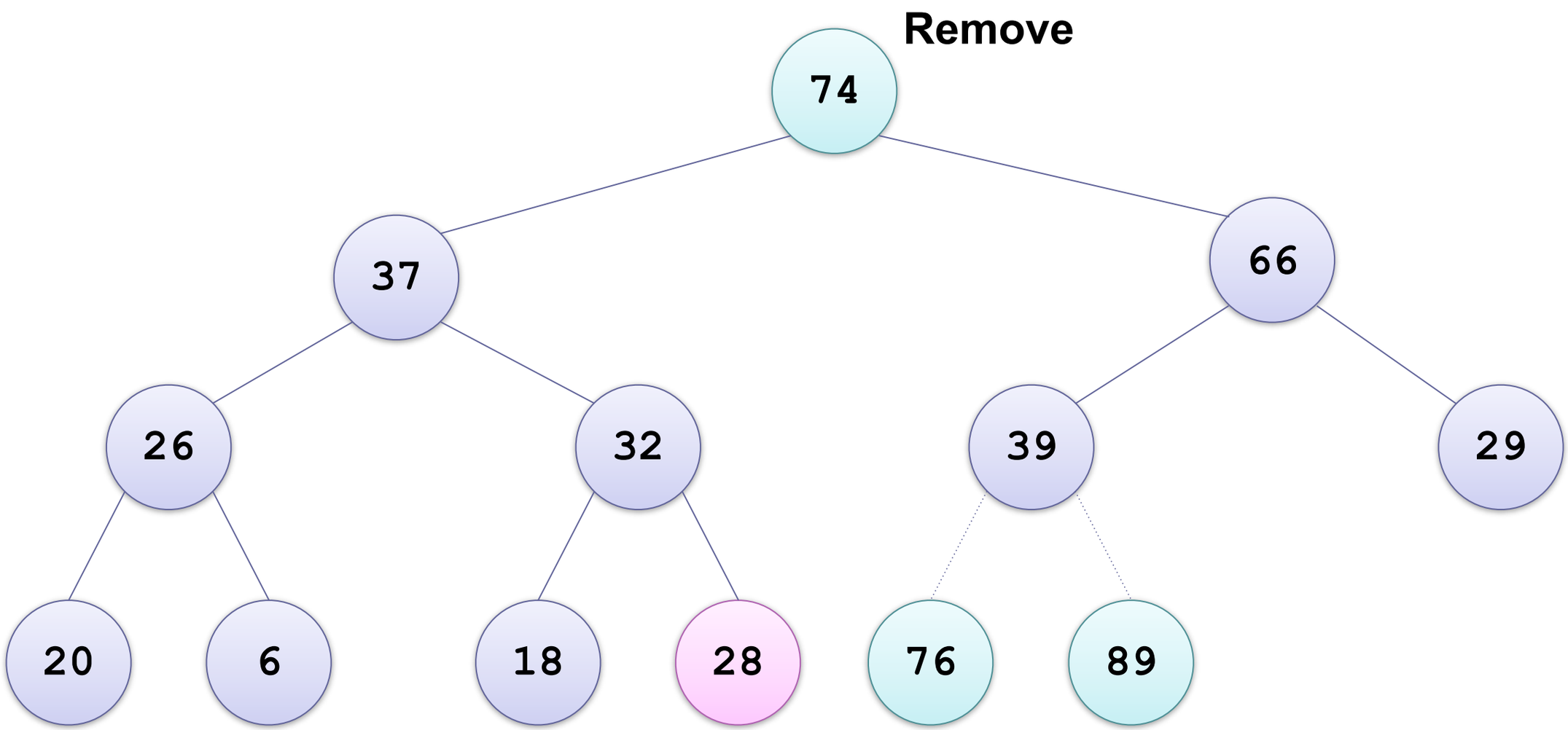
Trace of Heapsort (cont.)



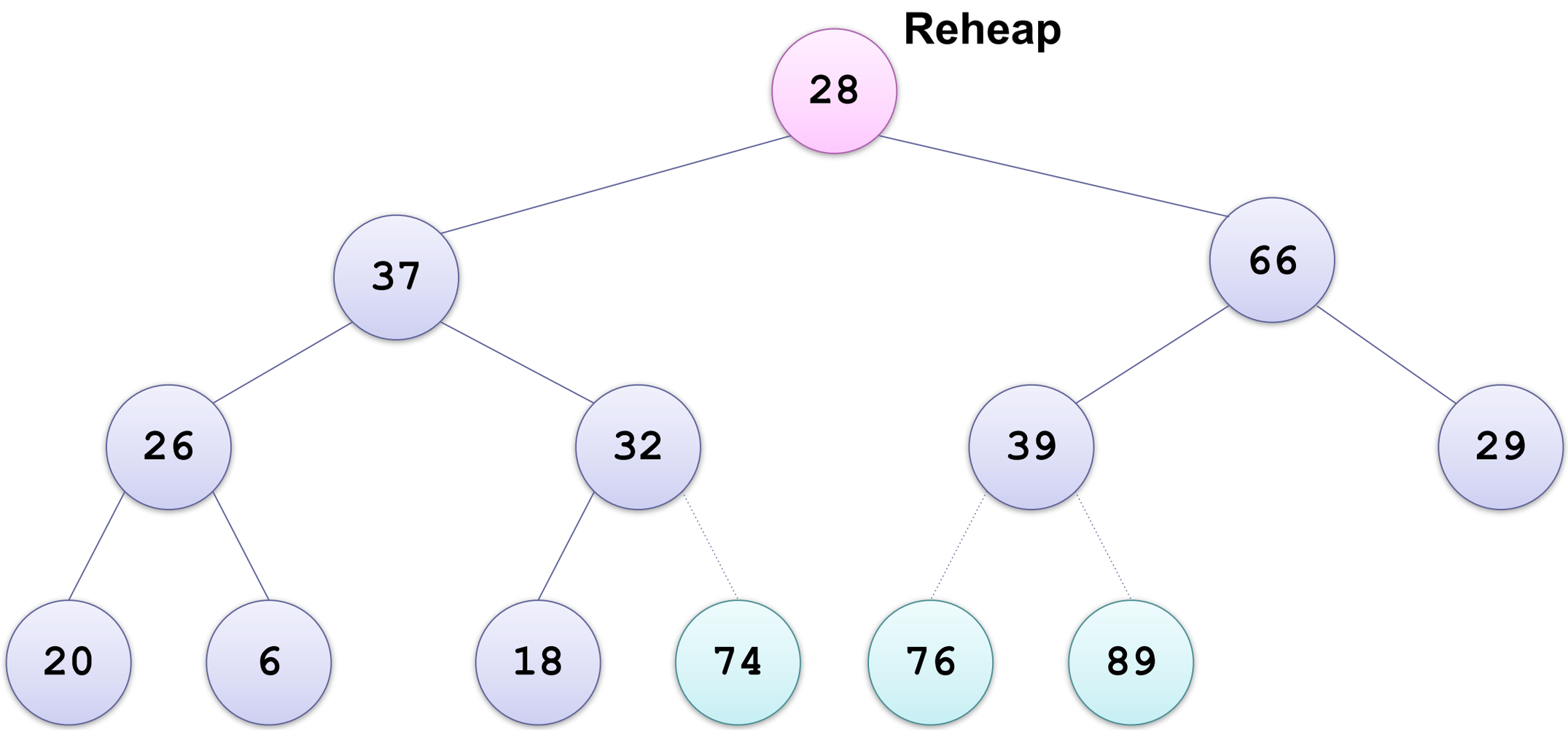
Trace of Heapsort (cont.)



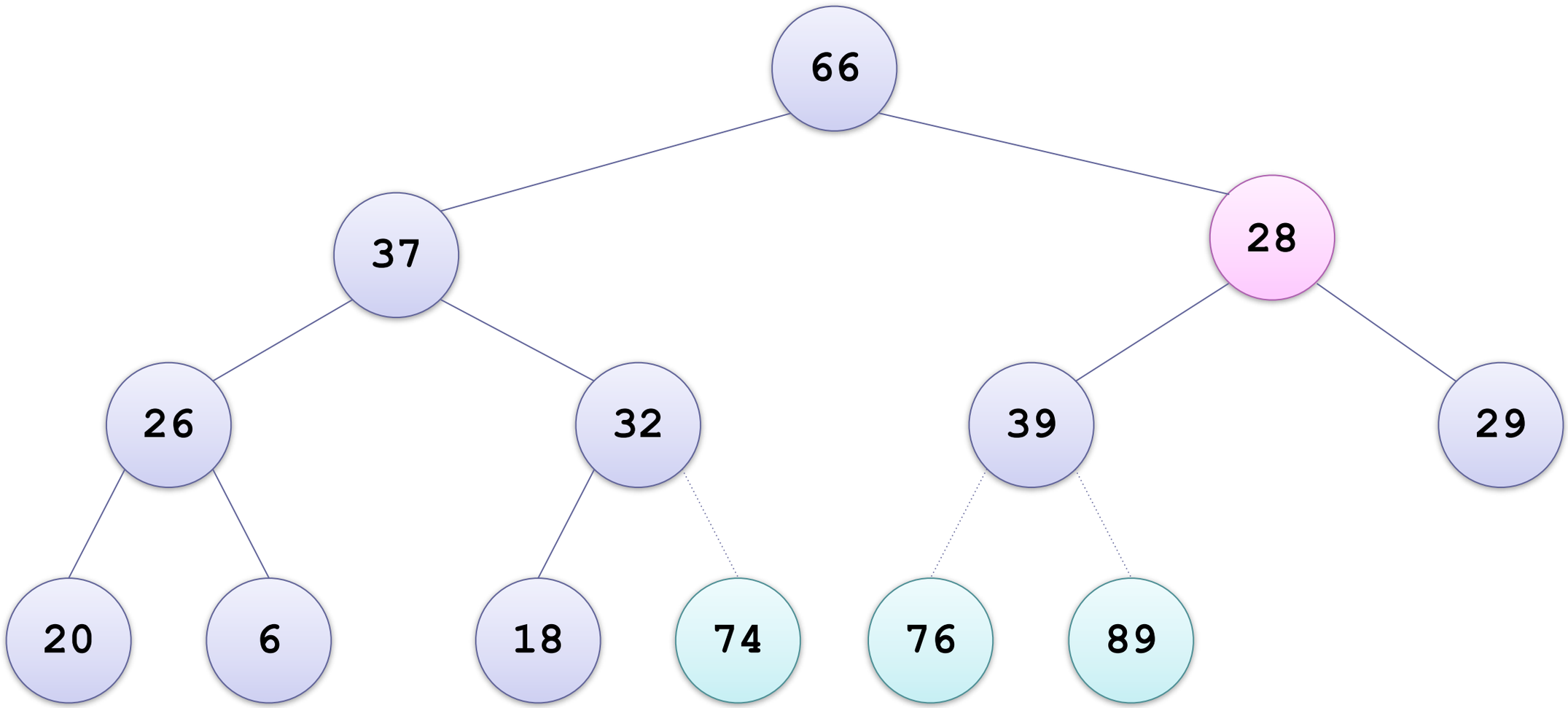
Trace of Heapsort (cont.)



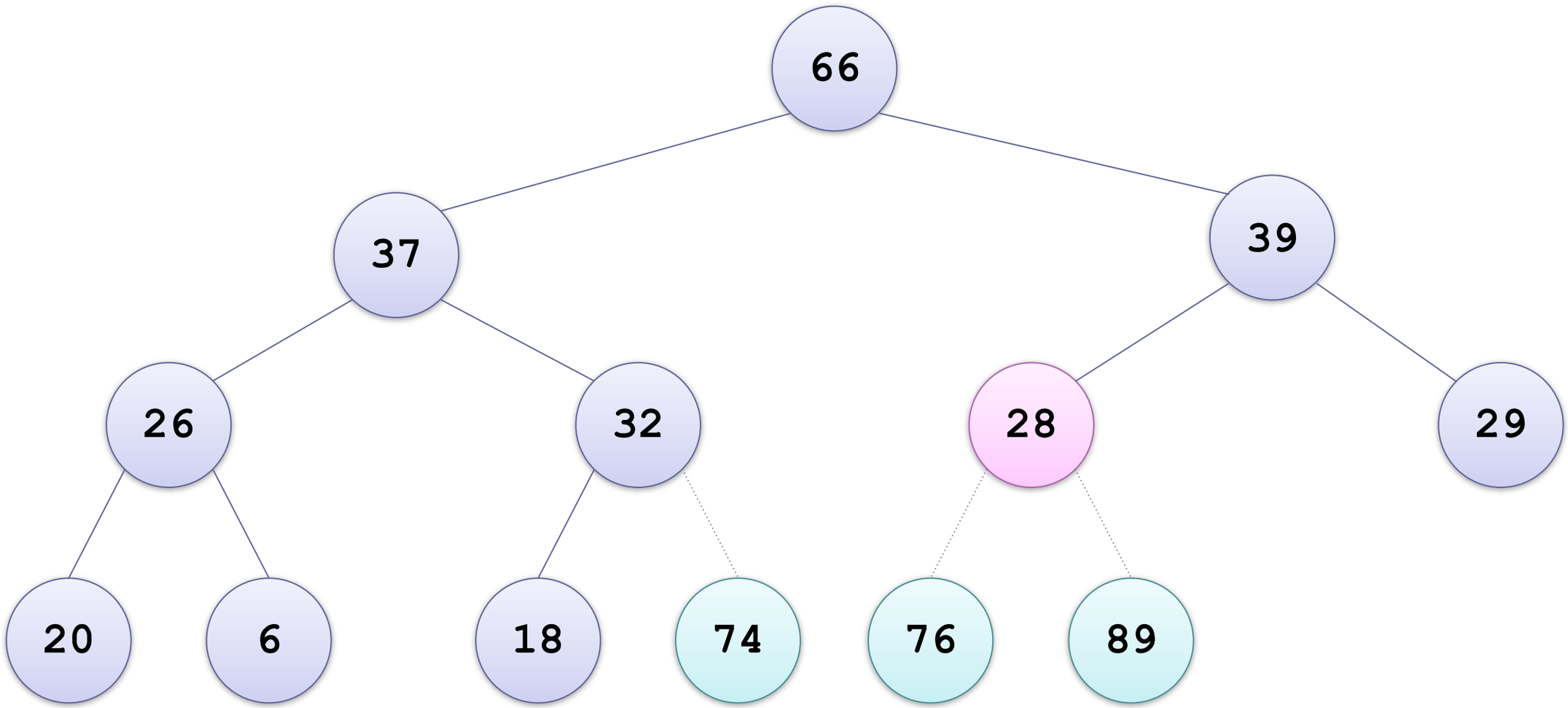
Trace of Heapsort (cont.)



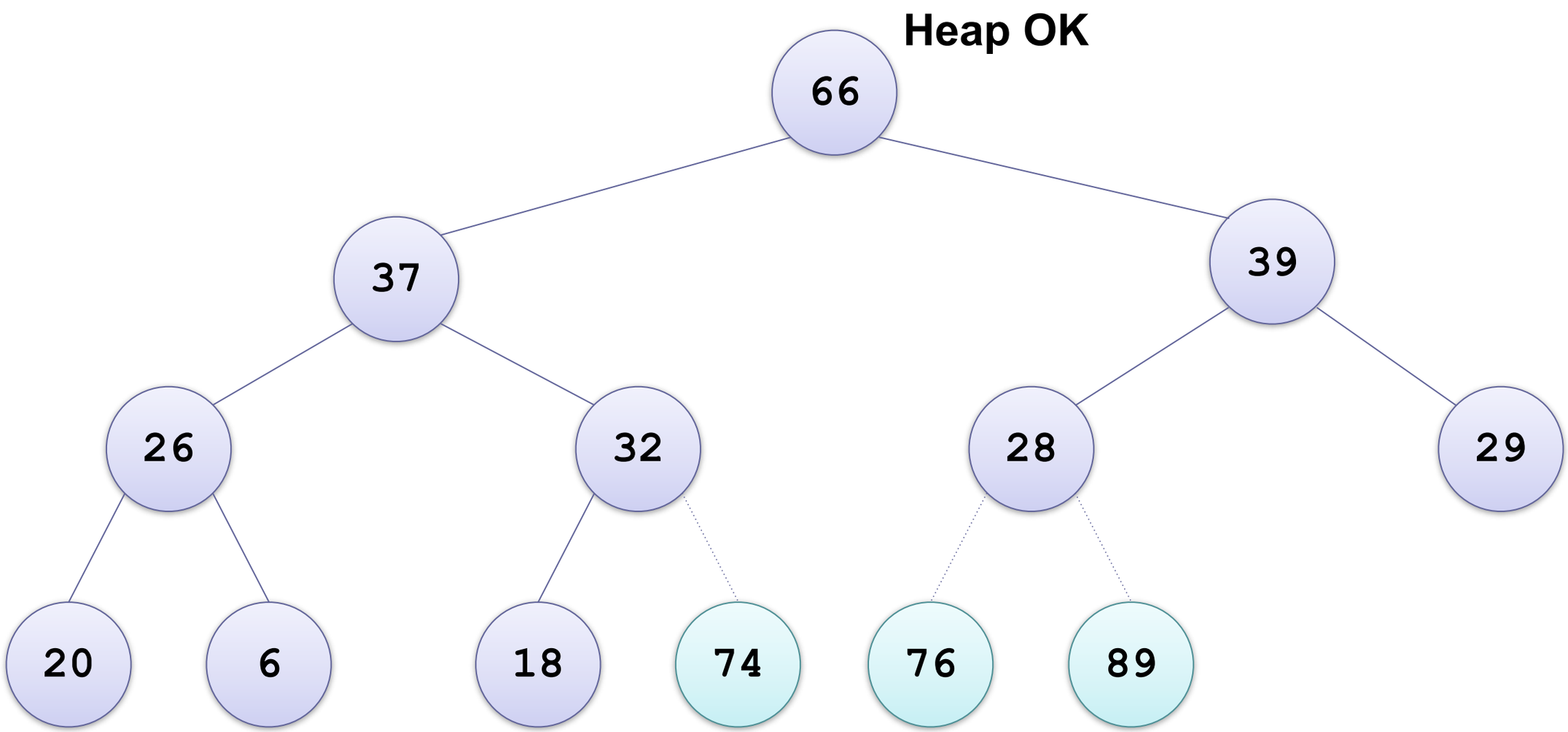
Trace of Heapsort (cont.)



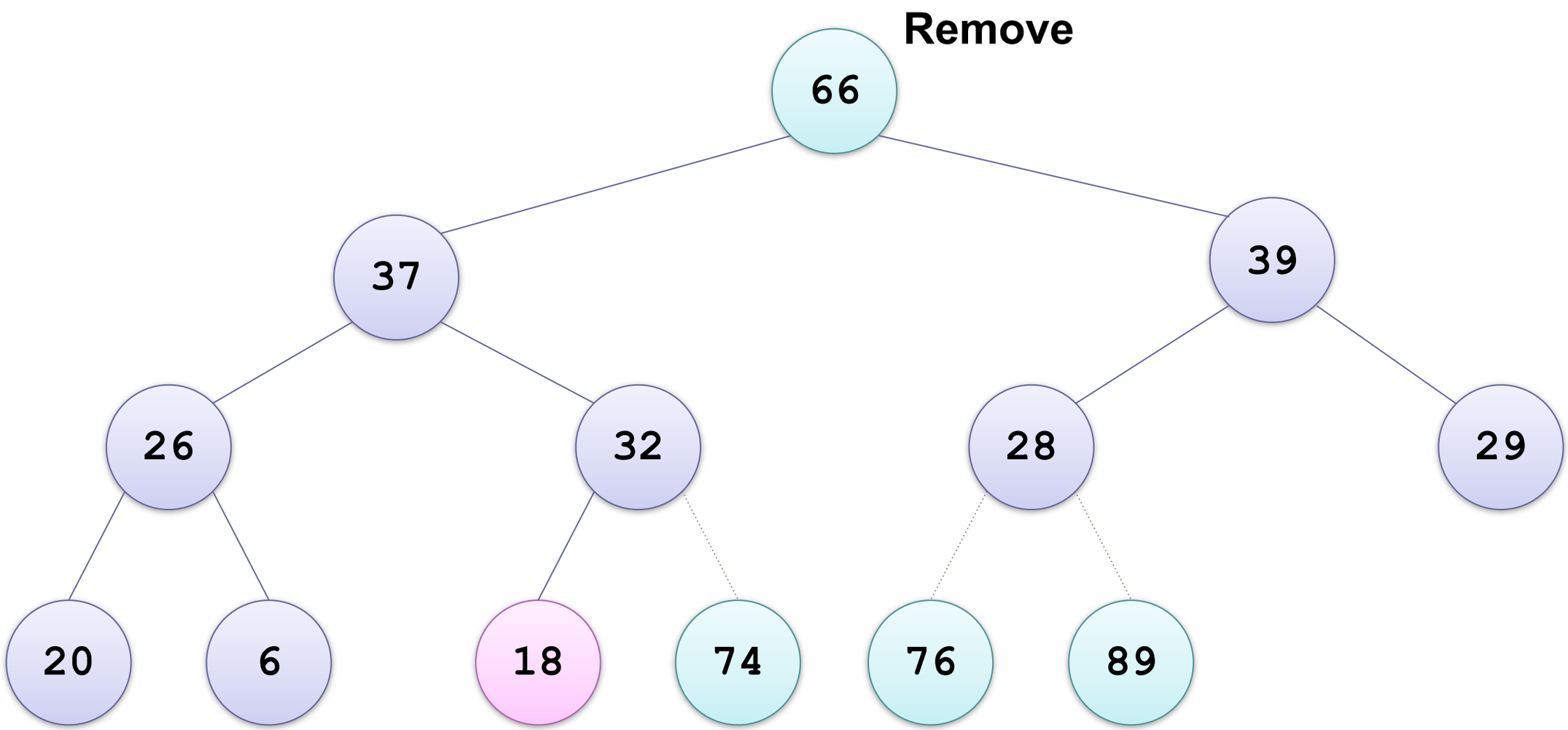
Trace of Heapsort (cont.)



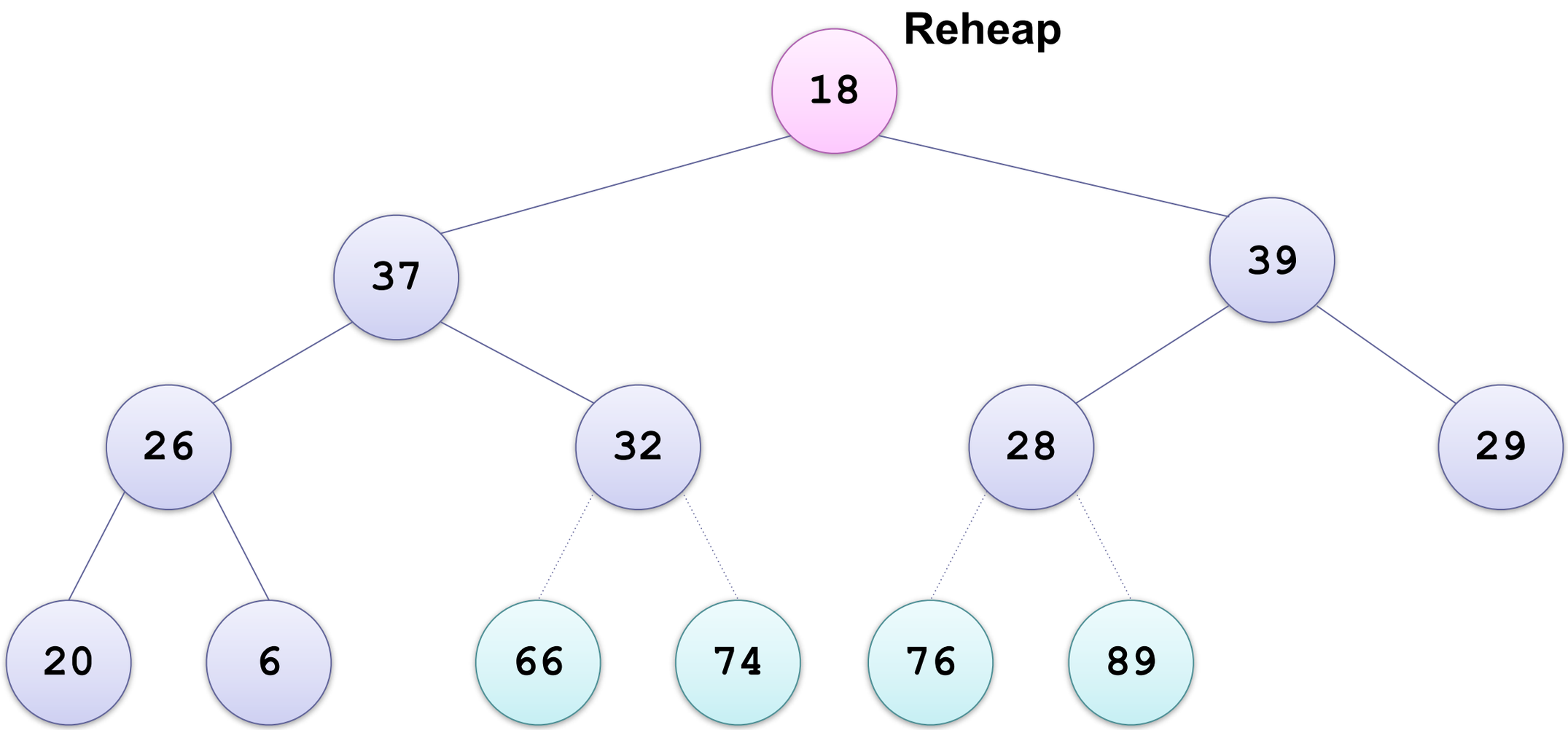
Trace of Heapsort (cont.)



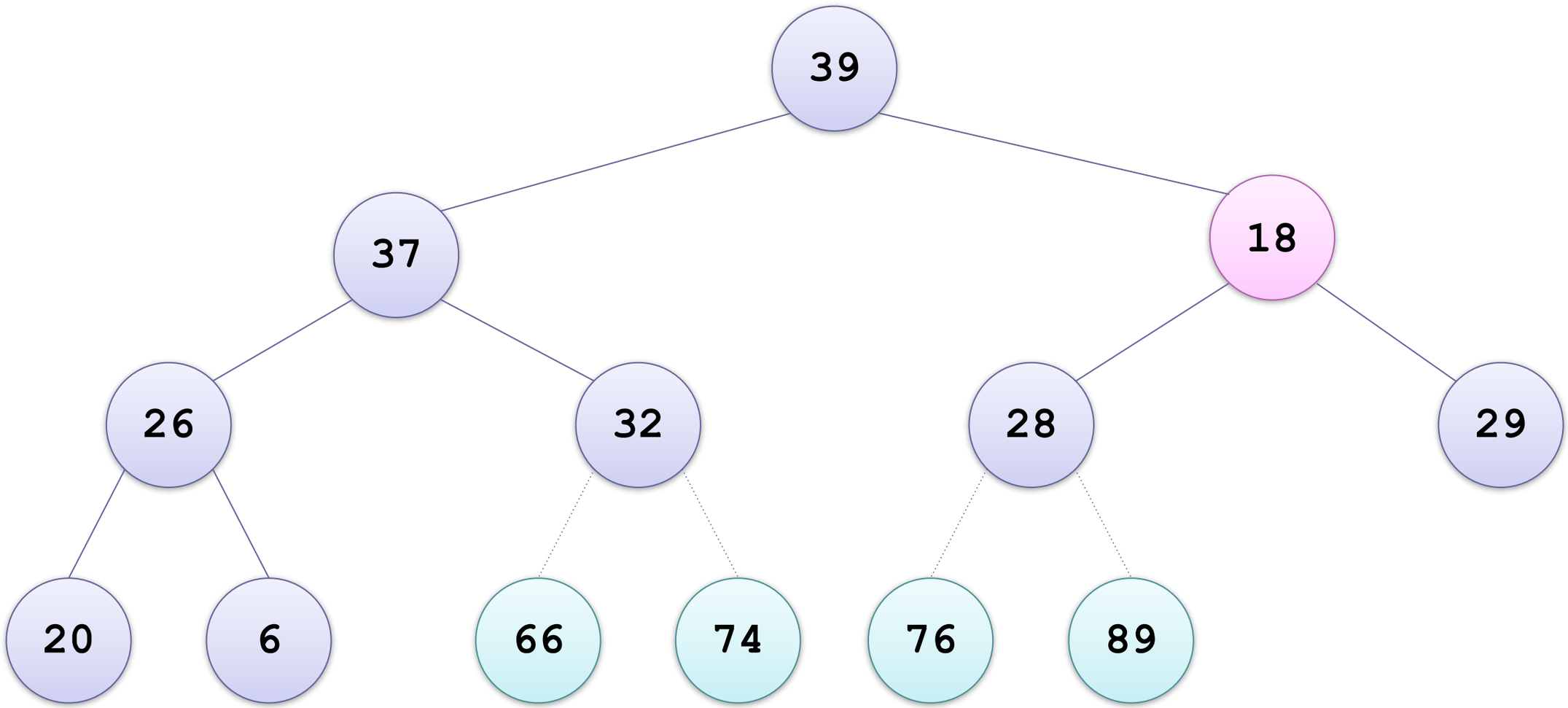
Trace of Heapsort (cont.)



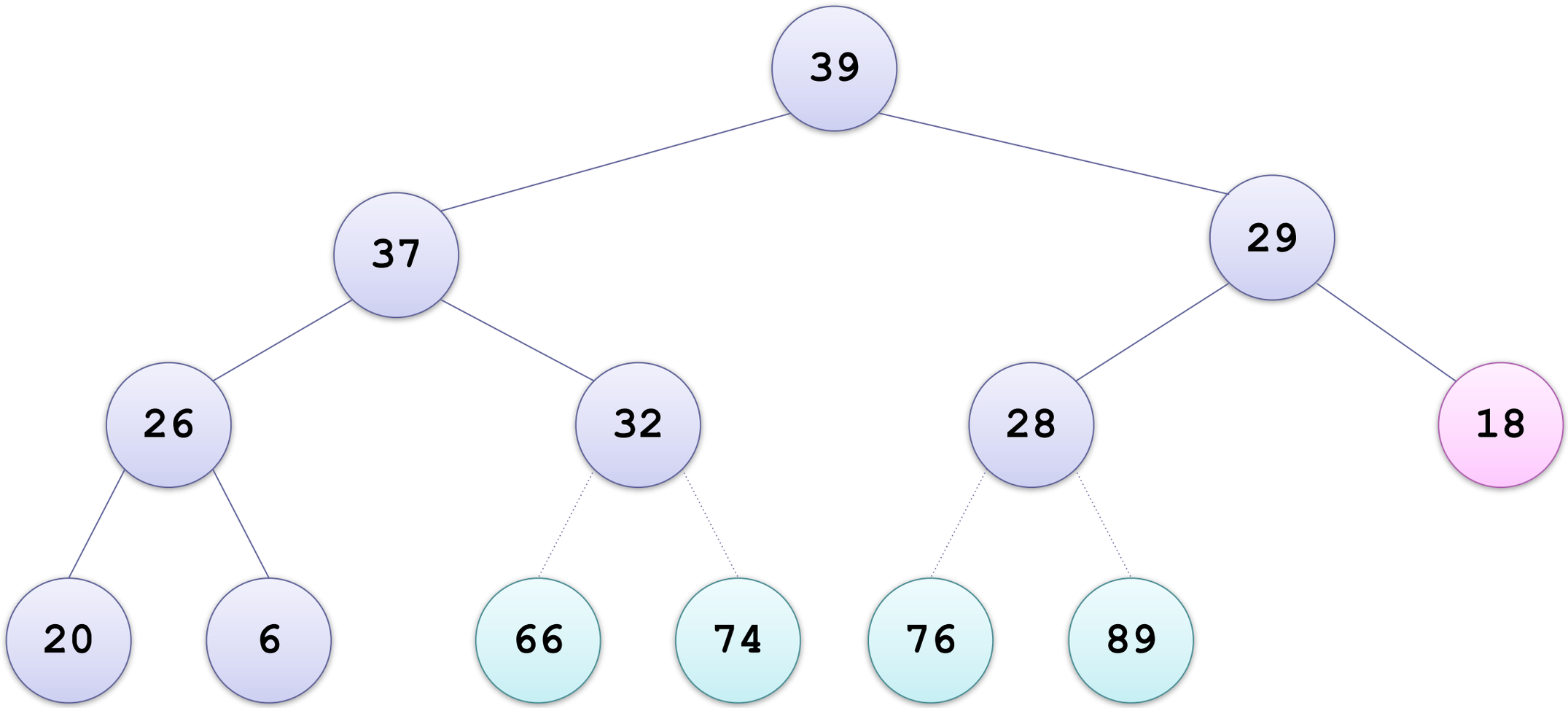
Trace of Heapsort (cont.)



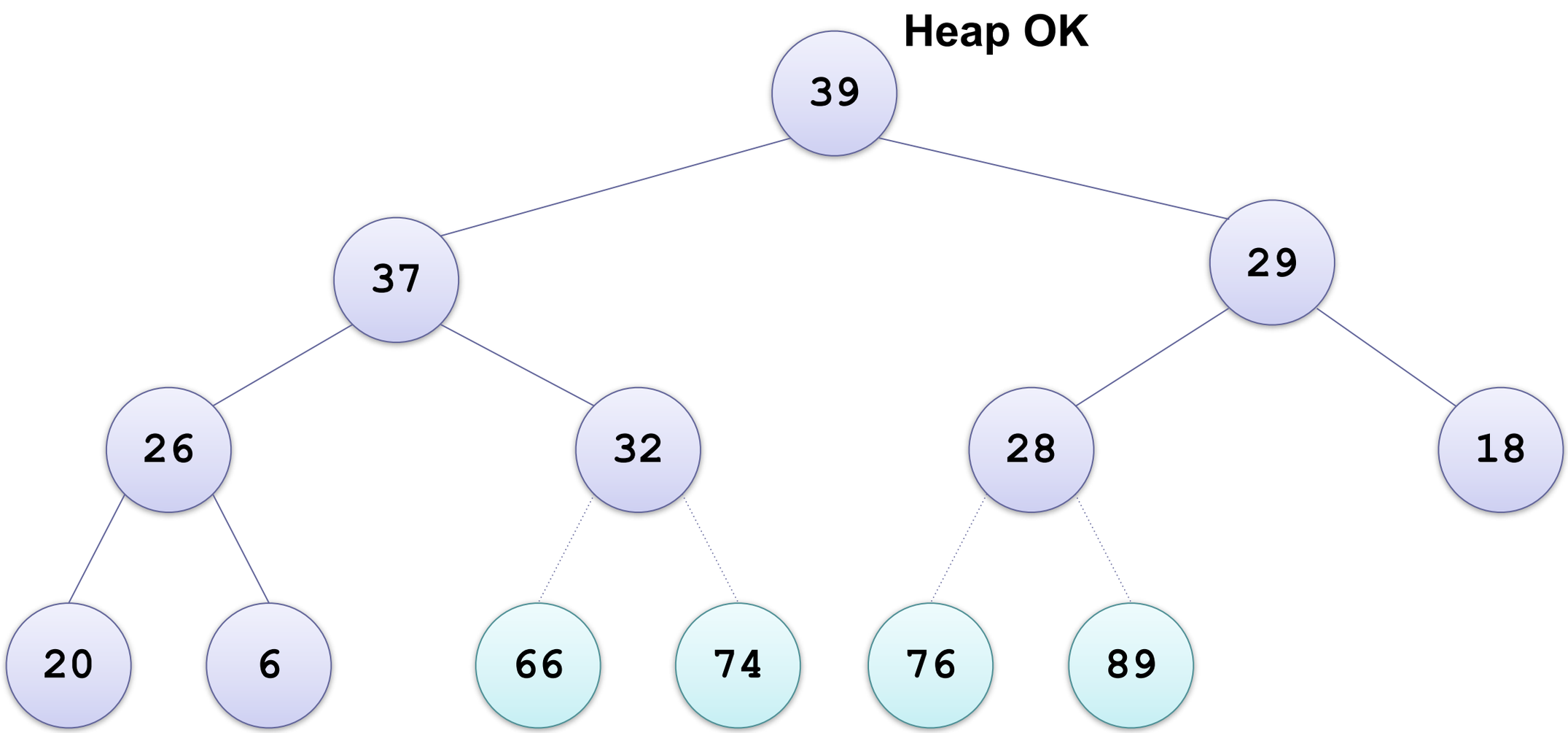
Trace of Heapsort (cont.)



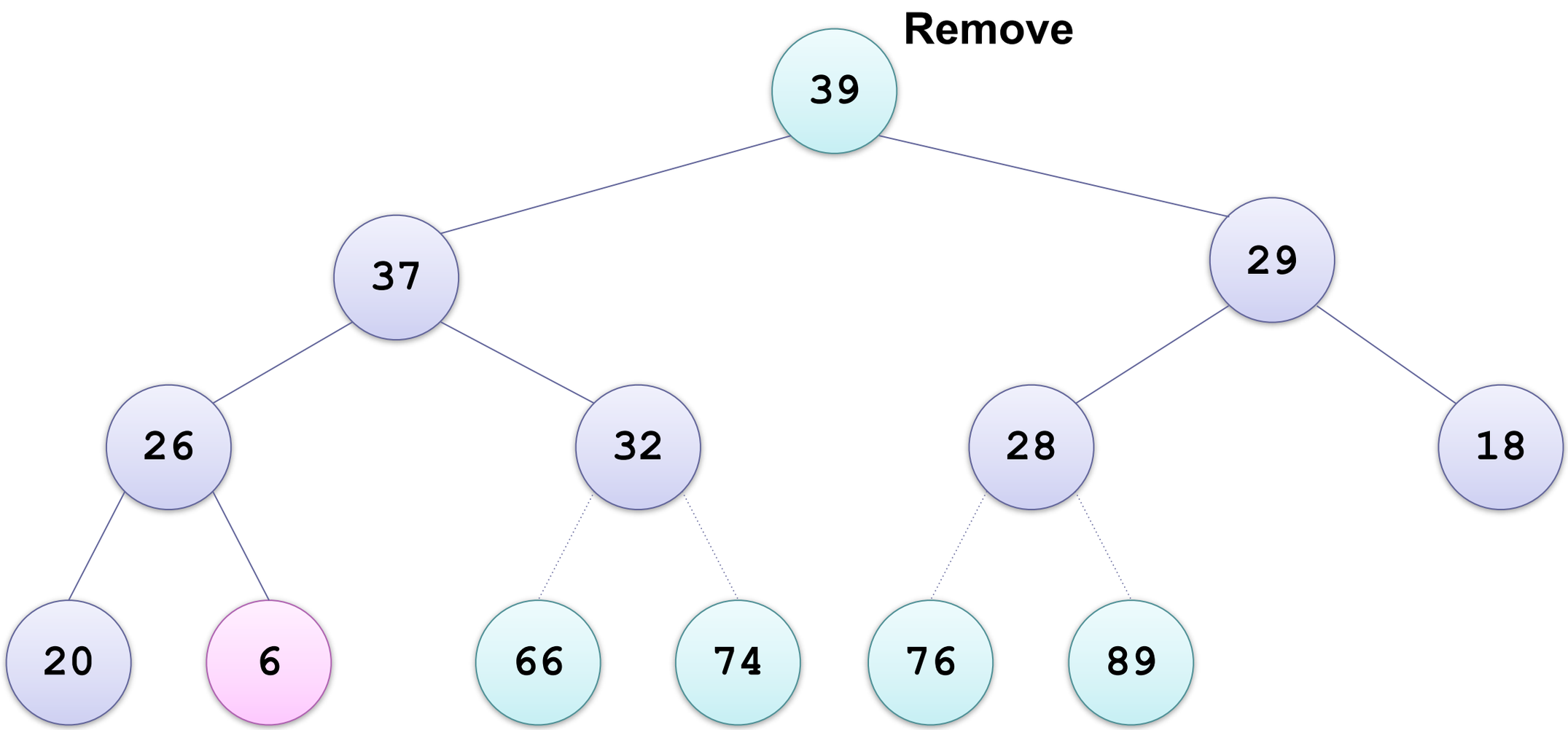
Trace of Heapsort (cont.)



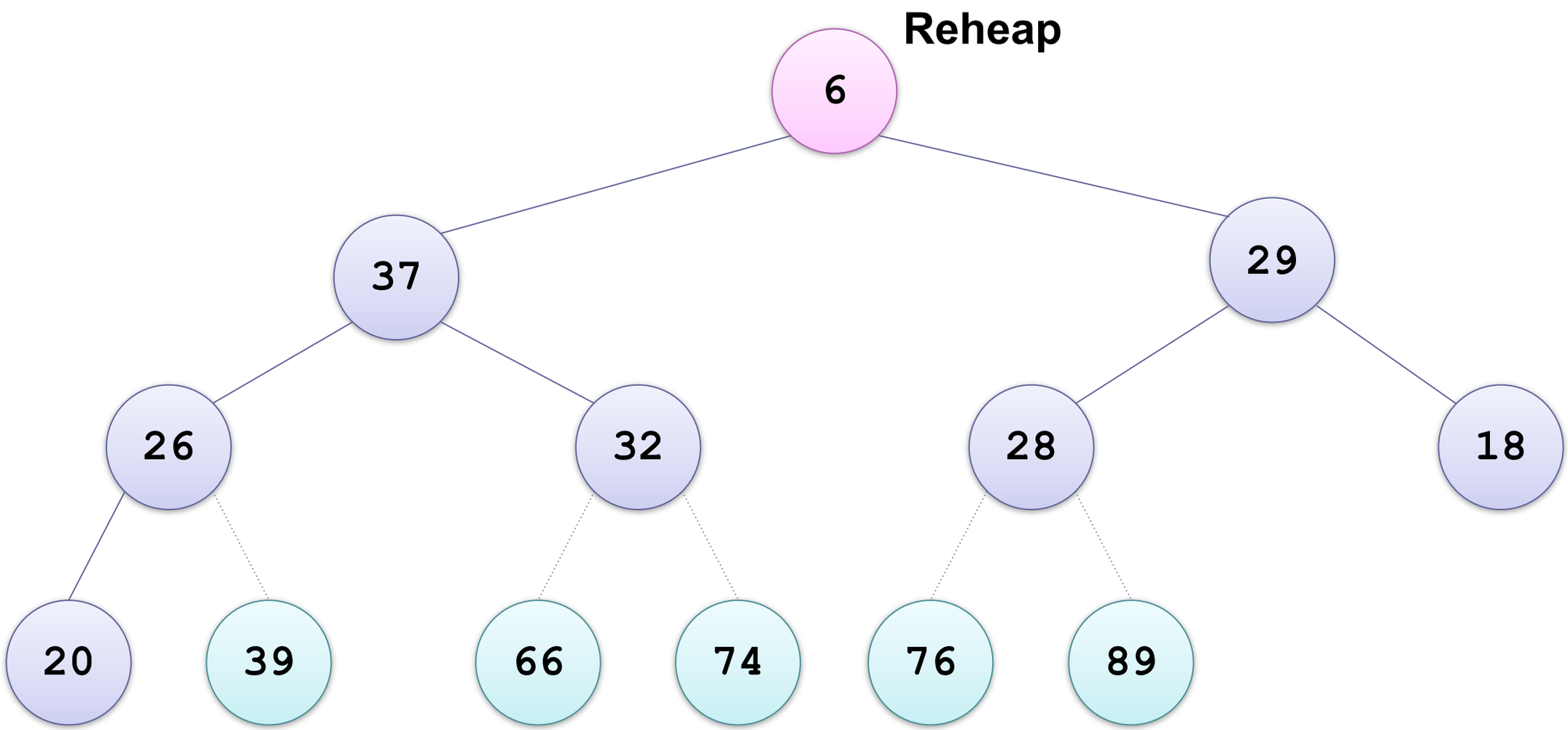
Trace of Heapsort (cont.)



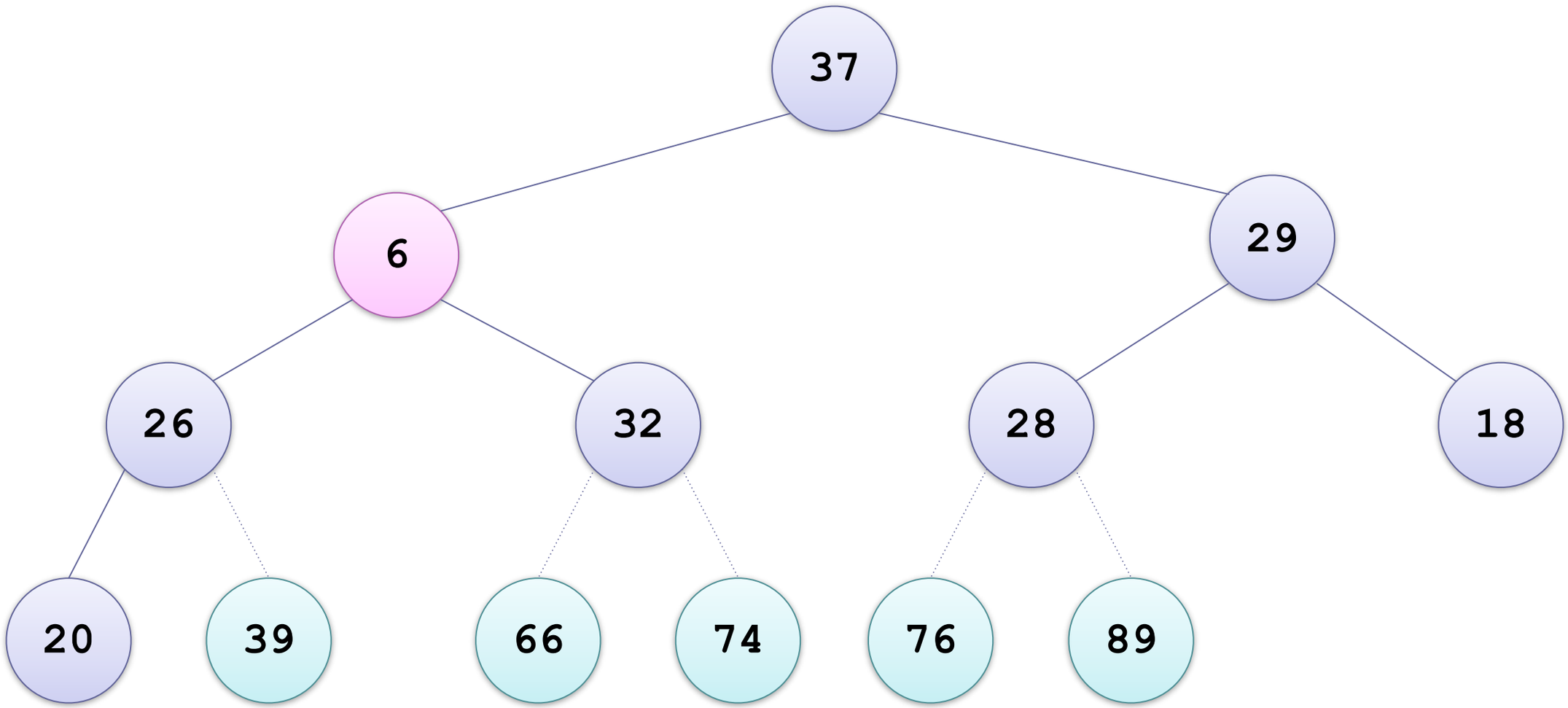
Trace of Heapsort (cont.)



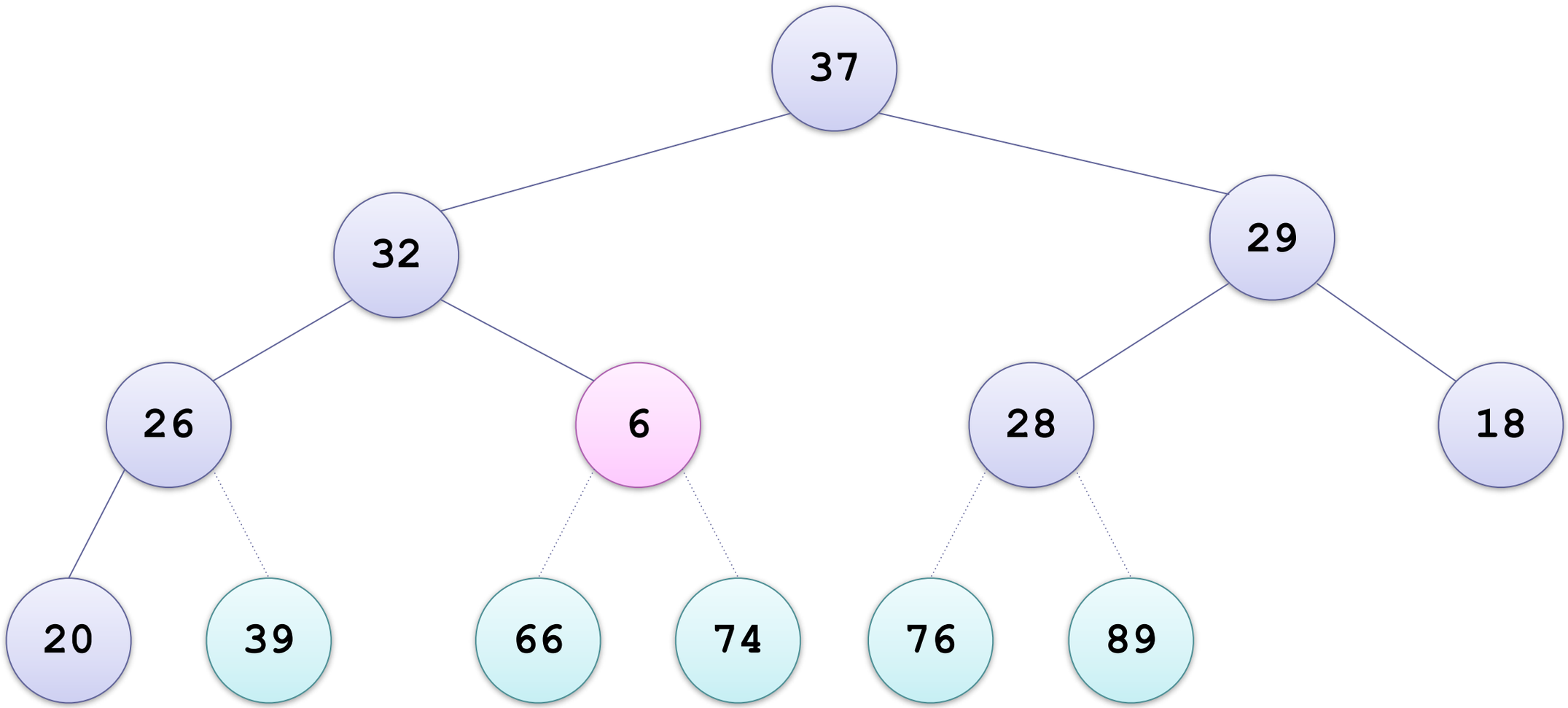
Trace of Heapsort (cont.)



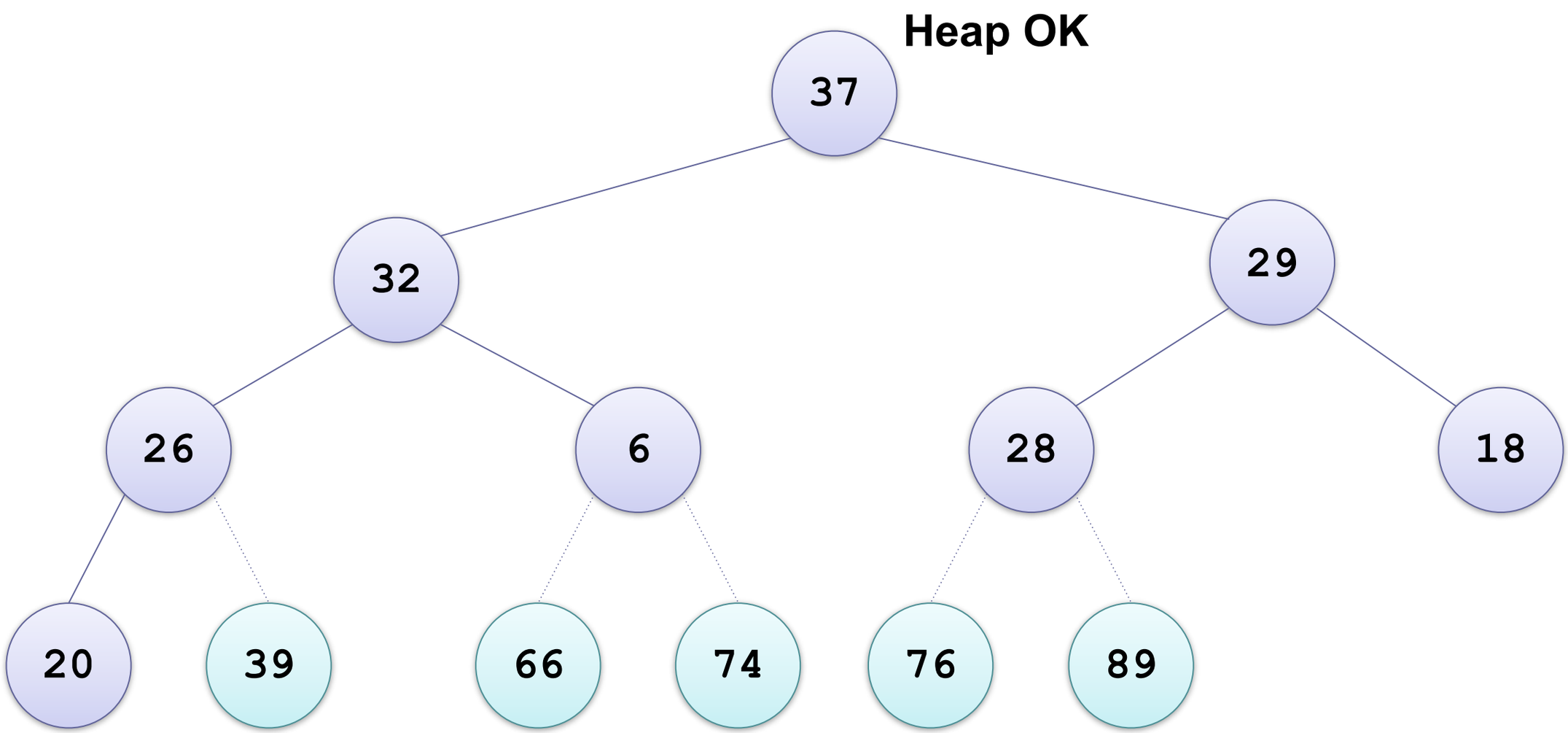
Trace of Heapsort (cont.)



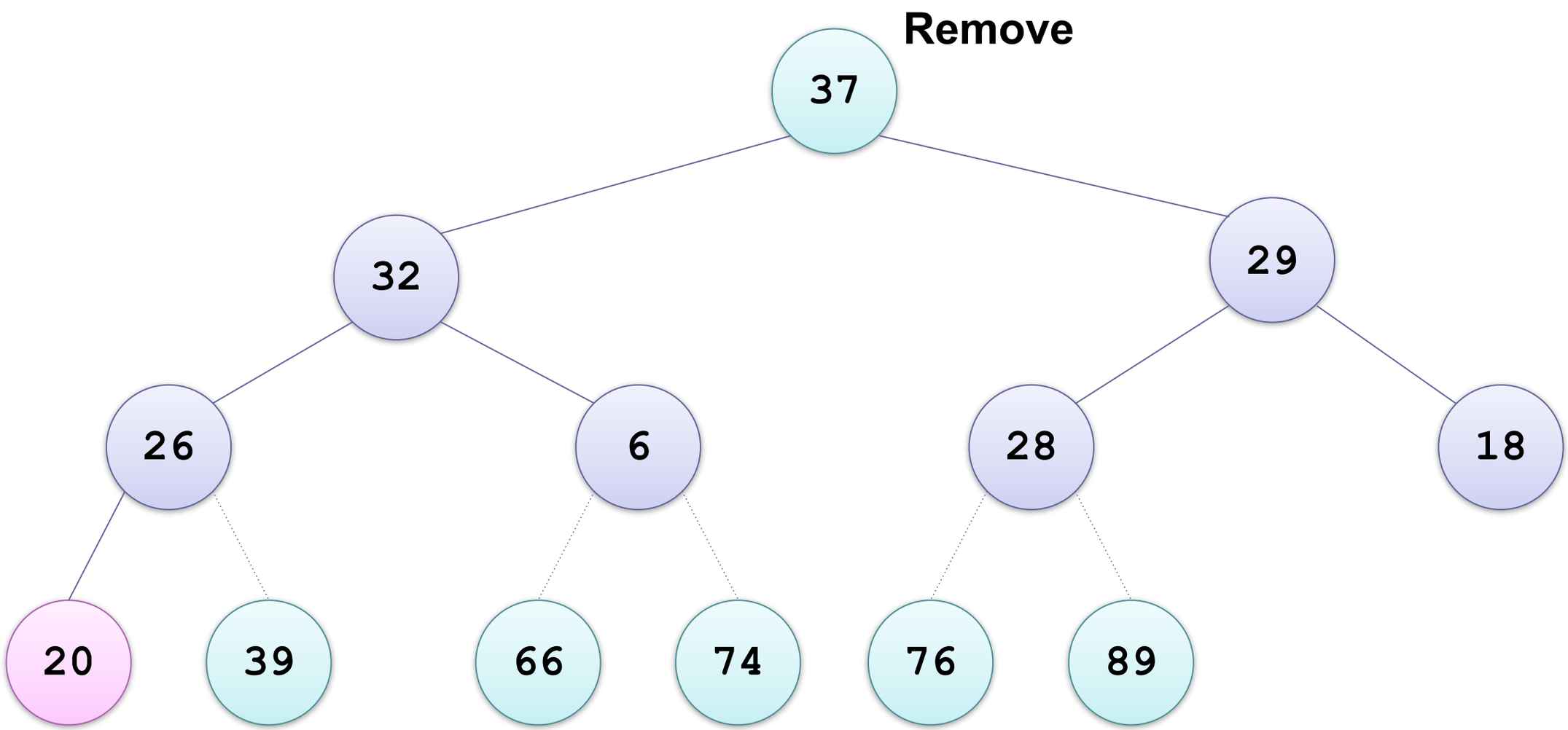
Trace of Heapsort (cont.)



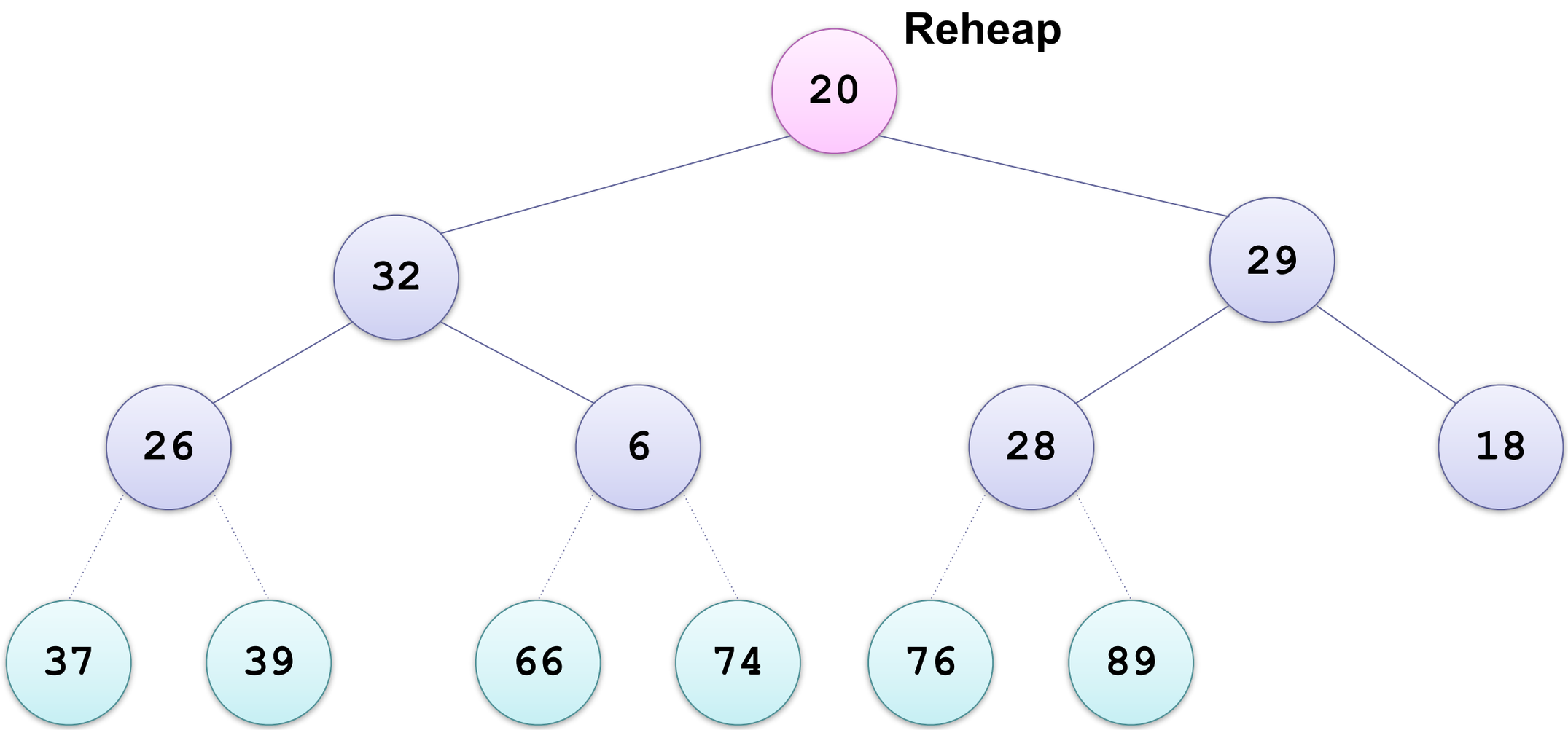
Trace of Heapsort (cont.)



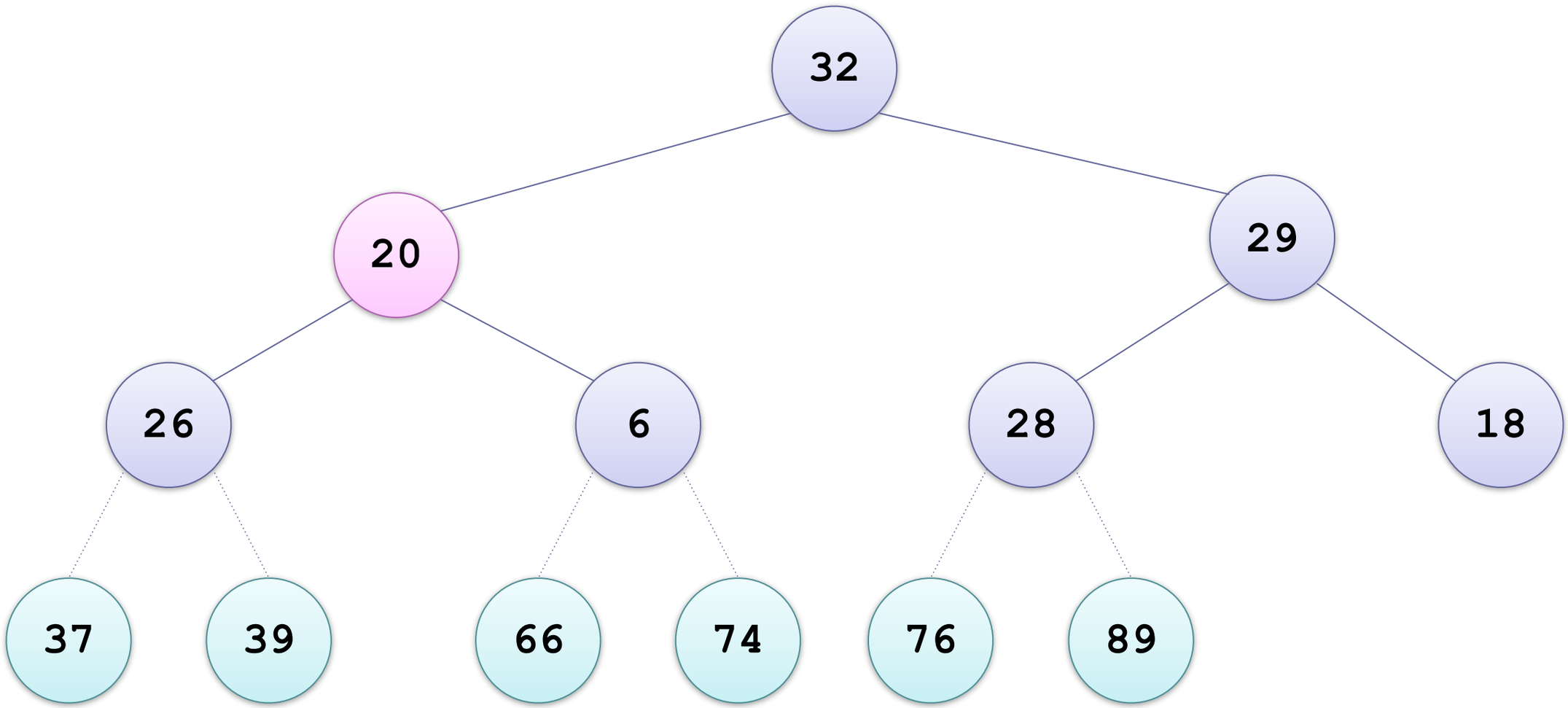
Trace of Heapsort (cont.)



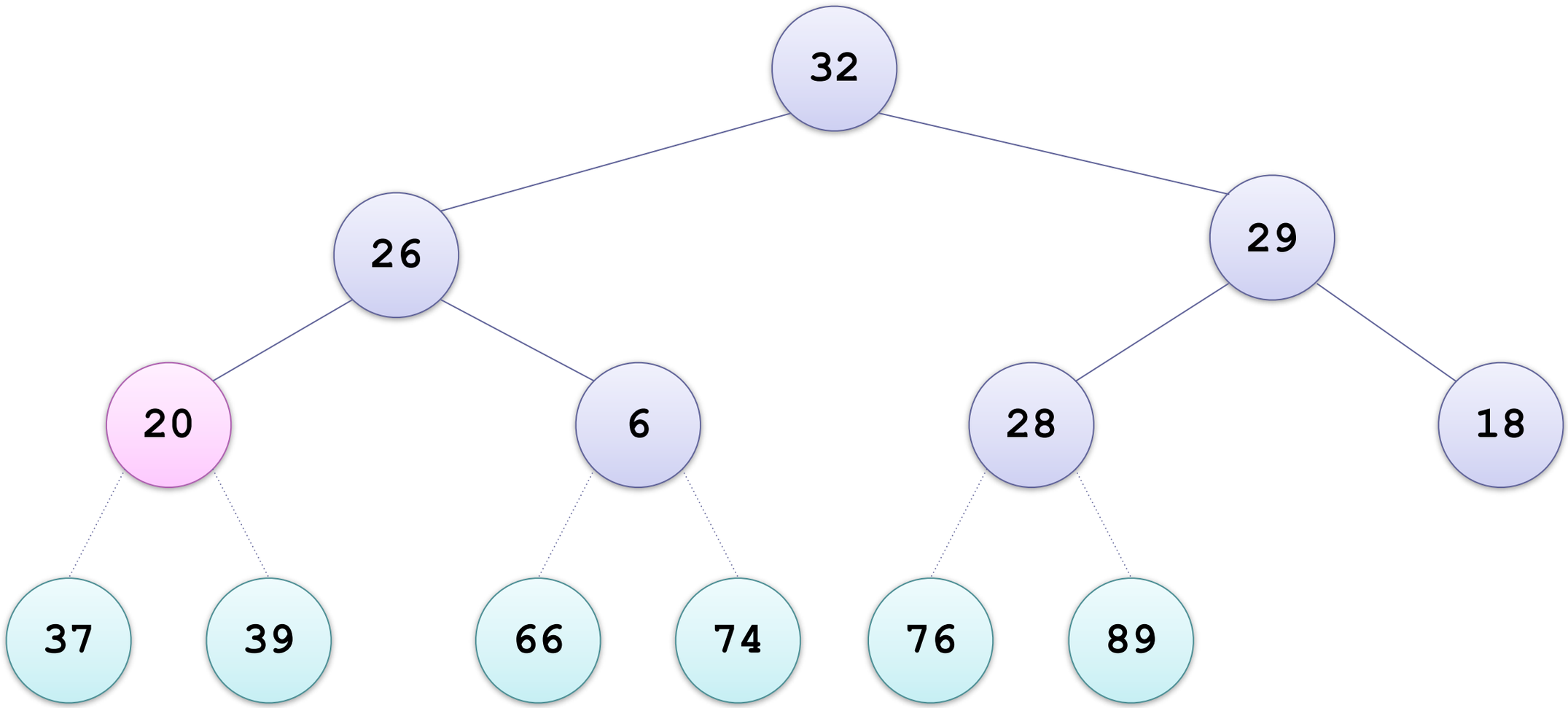
Trace of Heapsort (cont.)



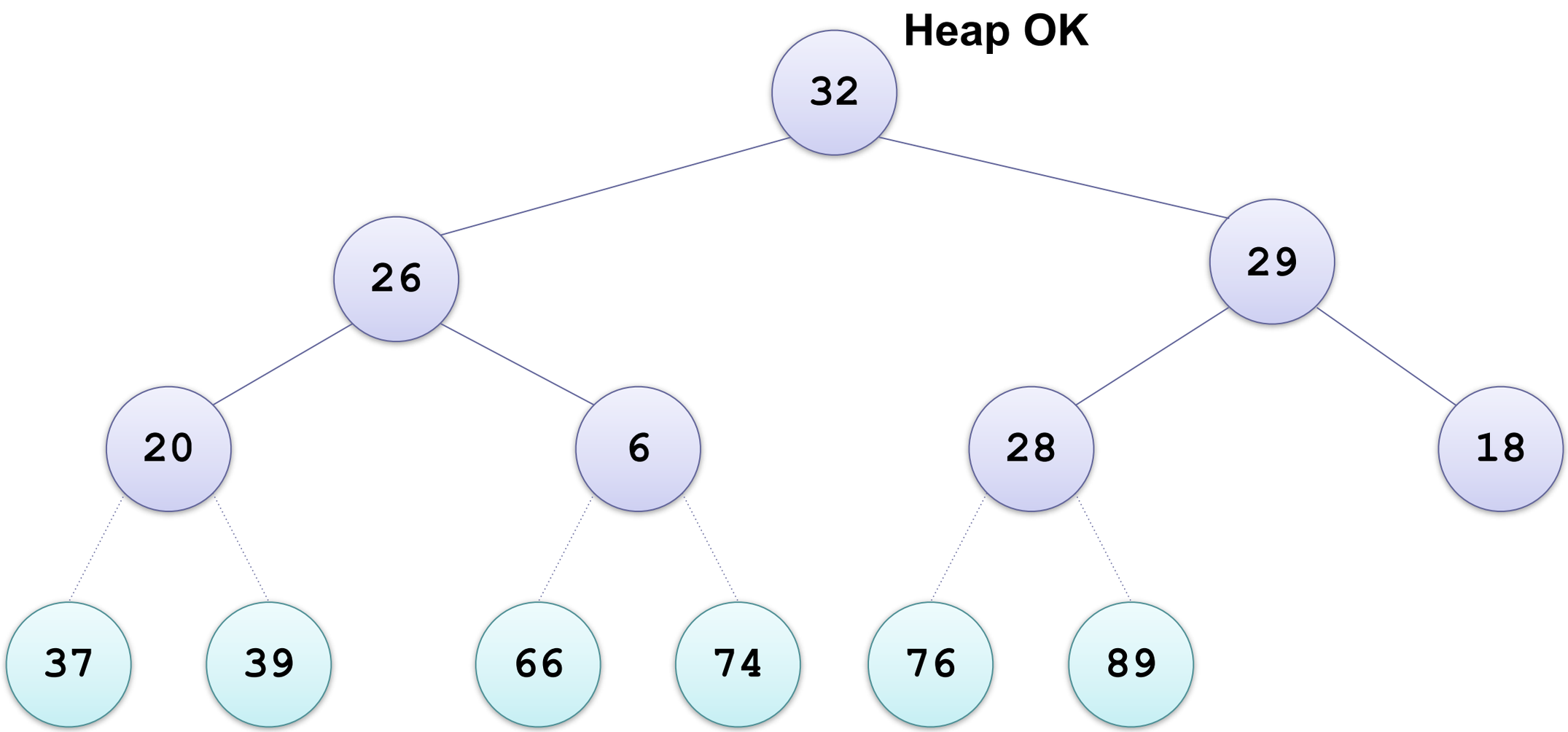
Trace of Heapsort (cont.)



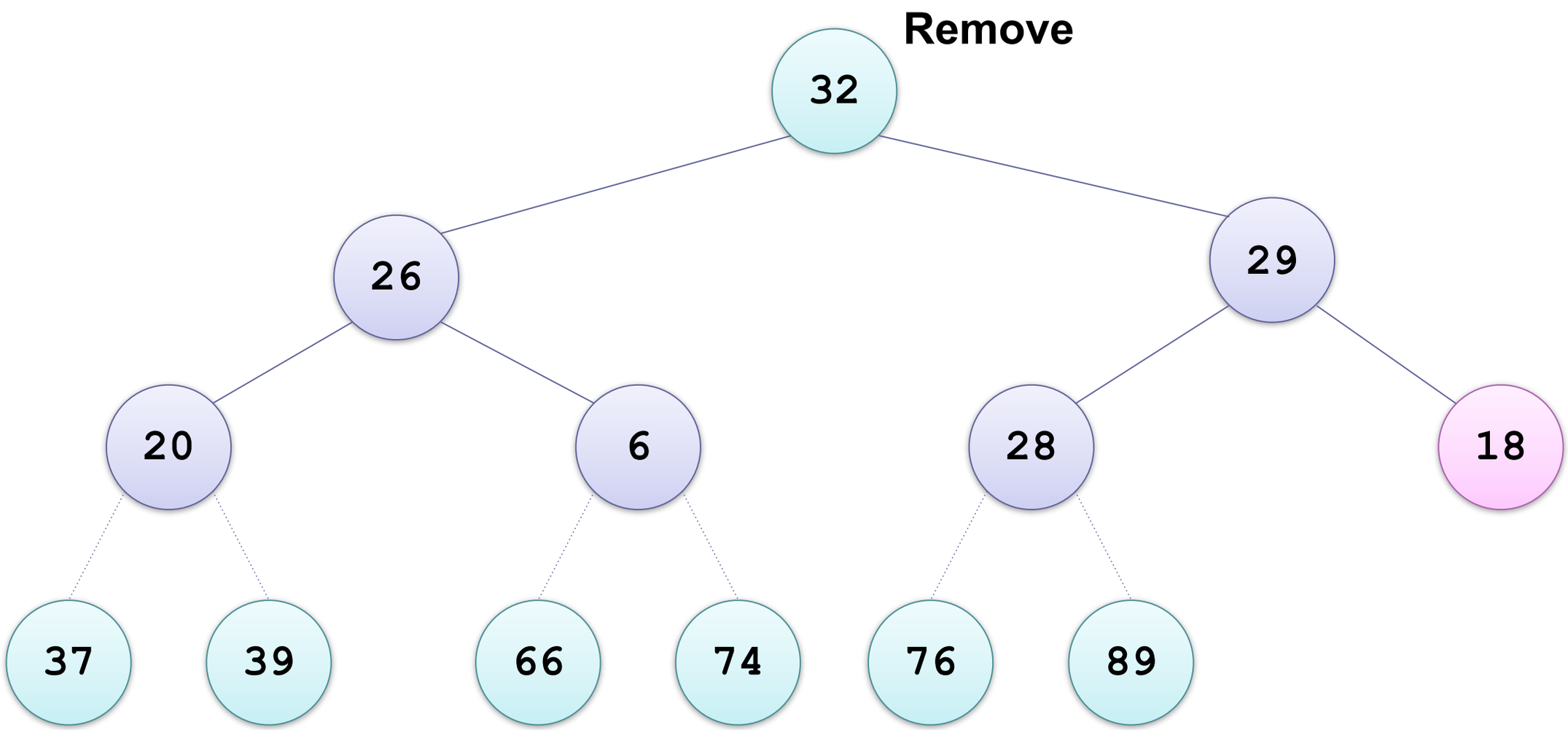
Trace of Heapsort (cont.)



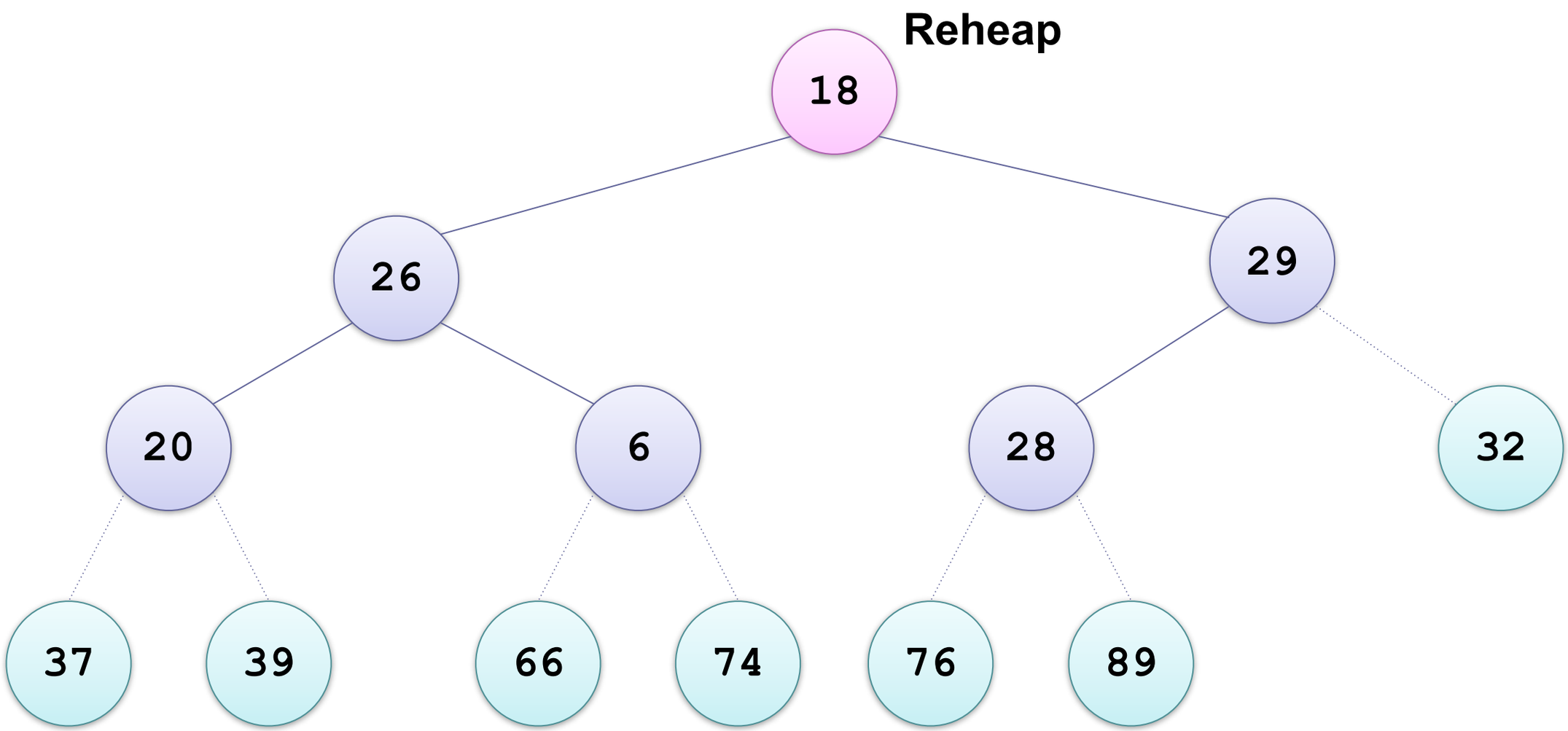
Trace of Heapsort (cont.)



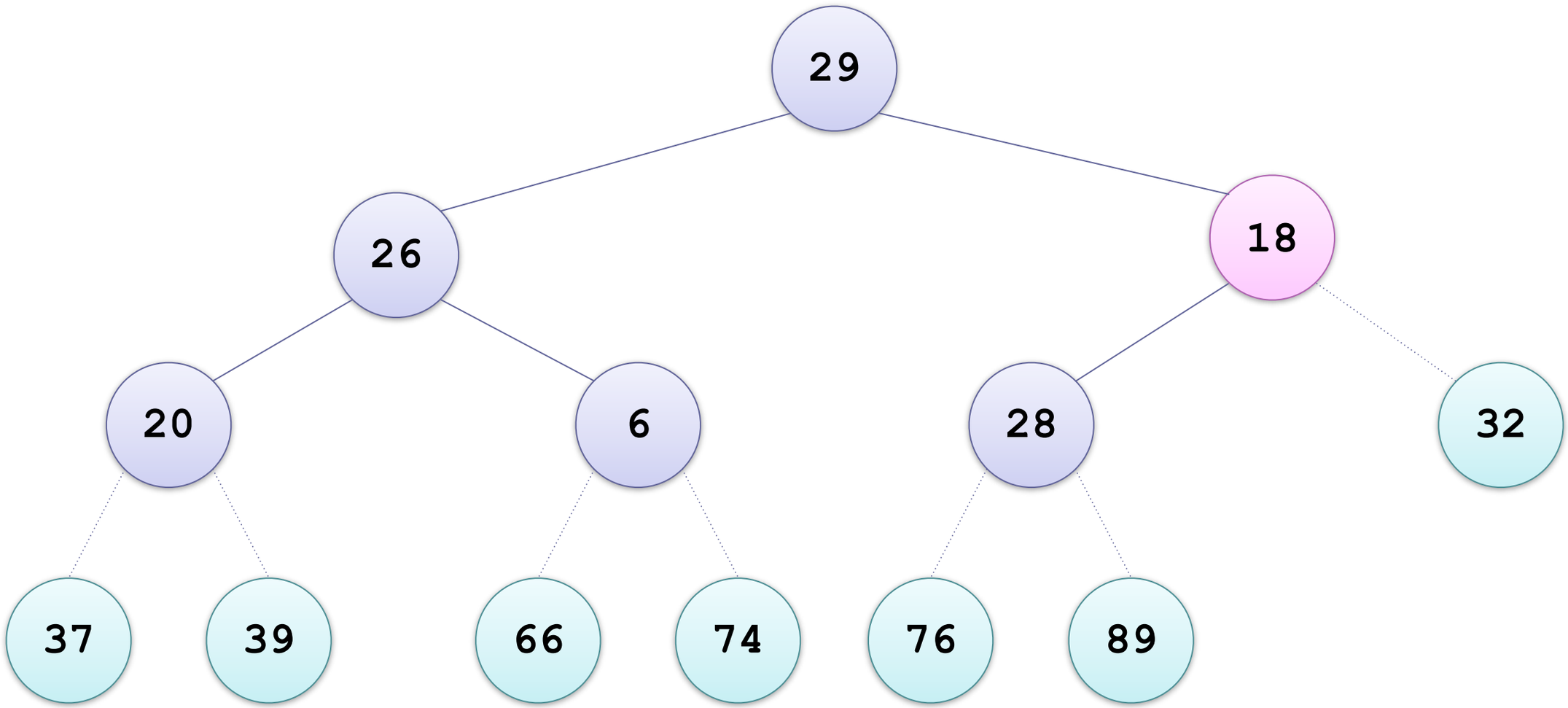
Trace of Heapsort (cont.)



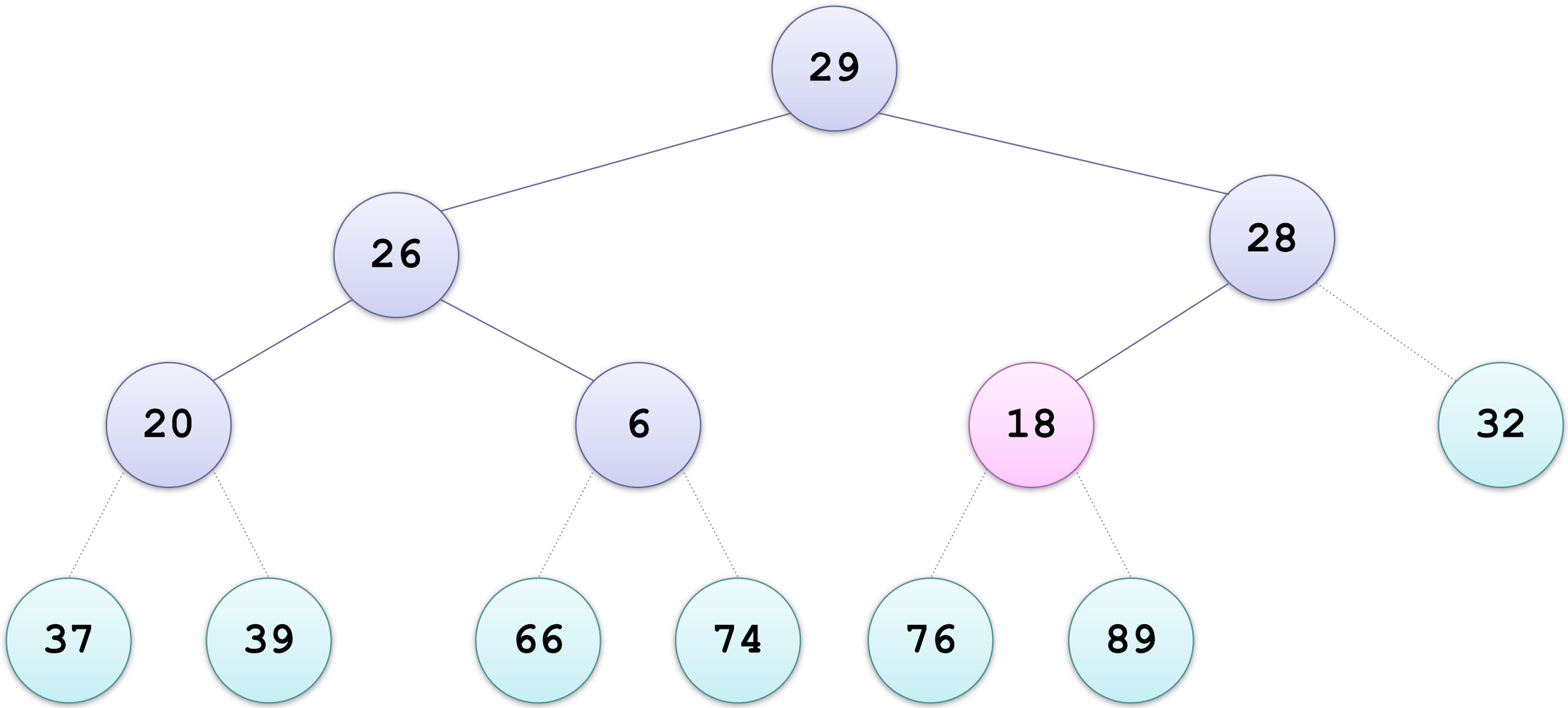
Trace of Heapsort (cont.)



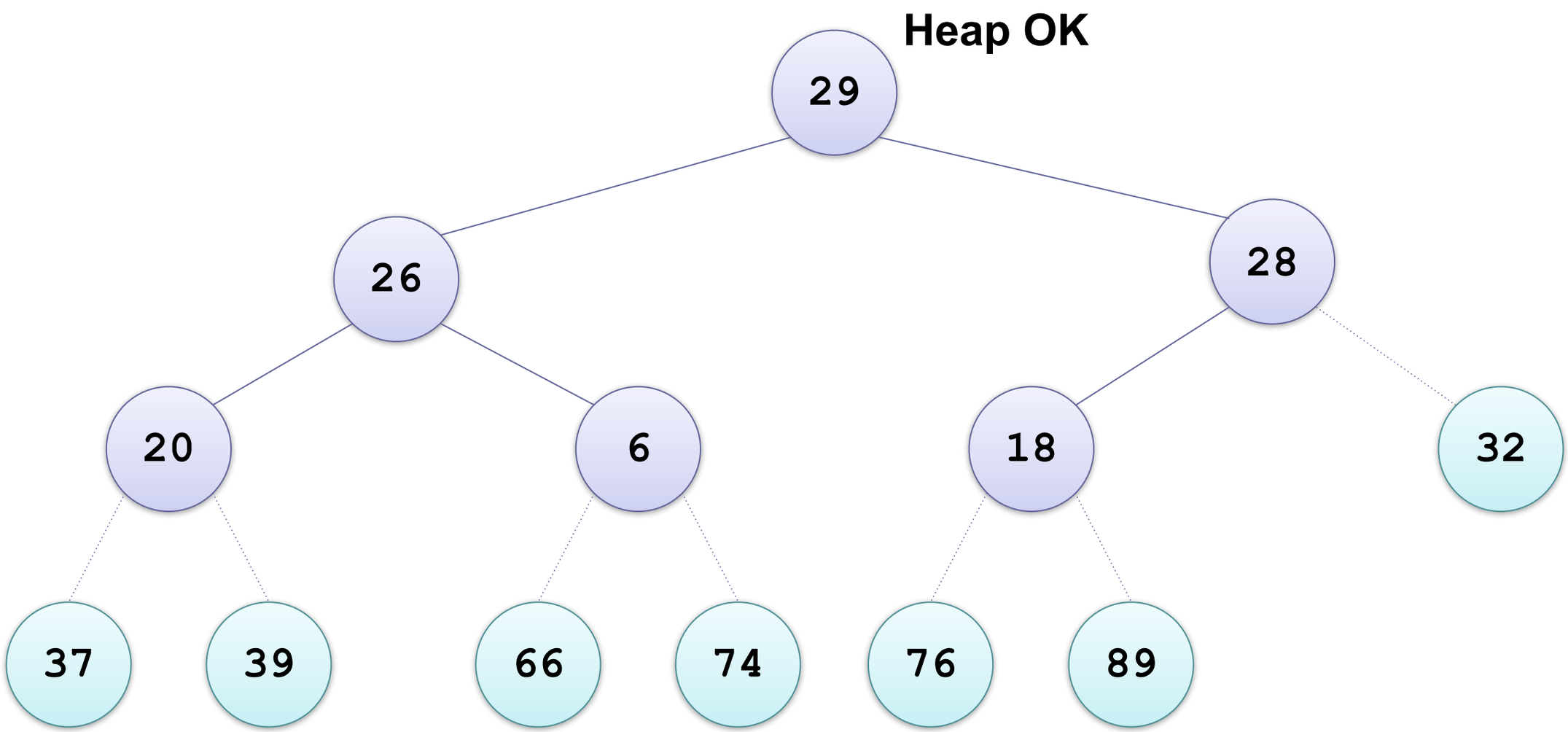
Trace of Heapsort (cont.)



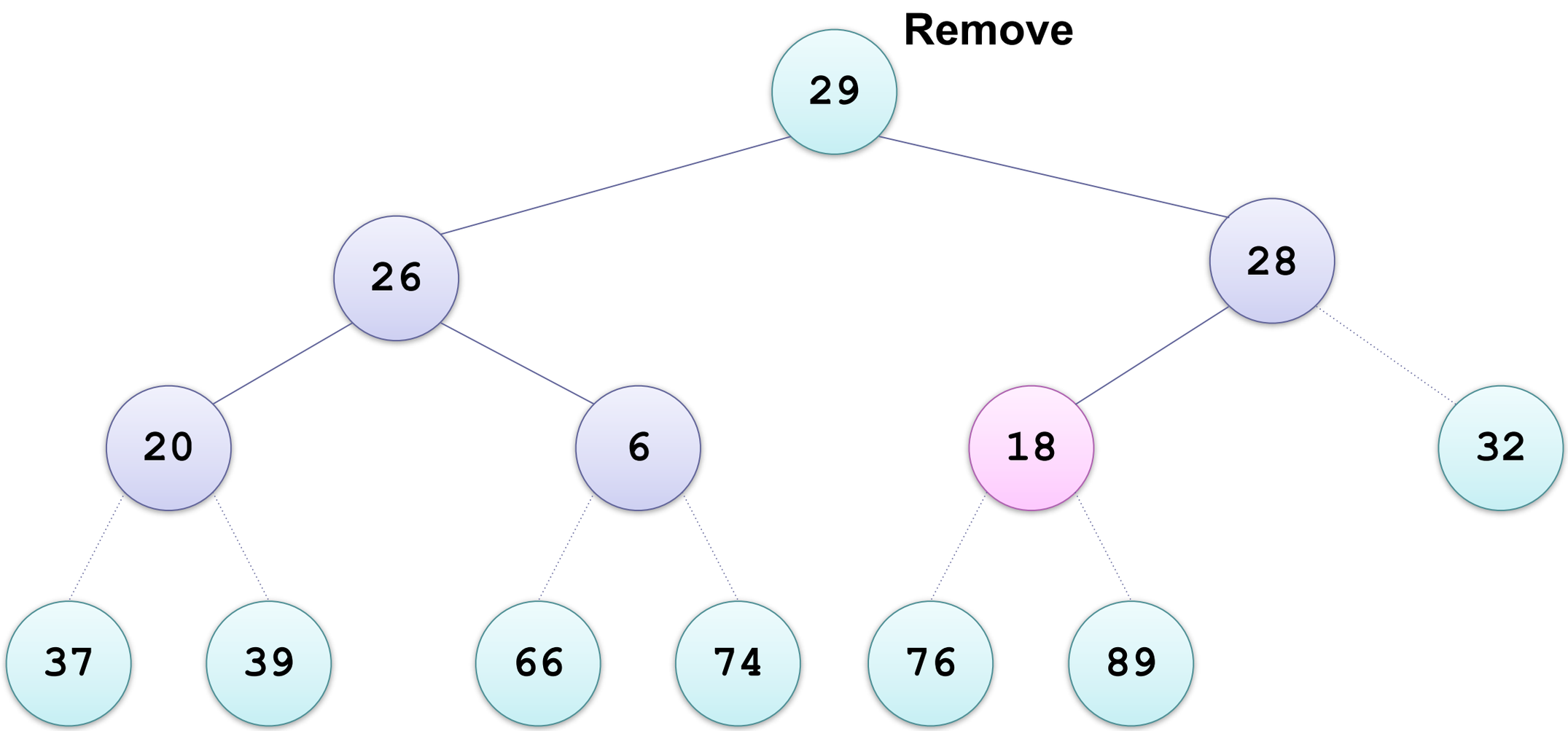
Trace of Heapsort (cont.)



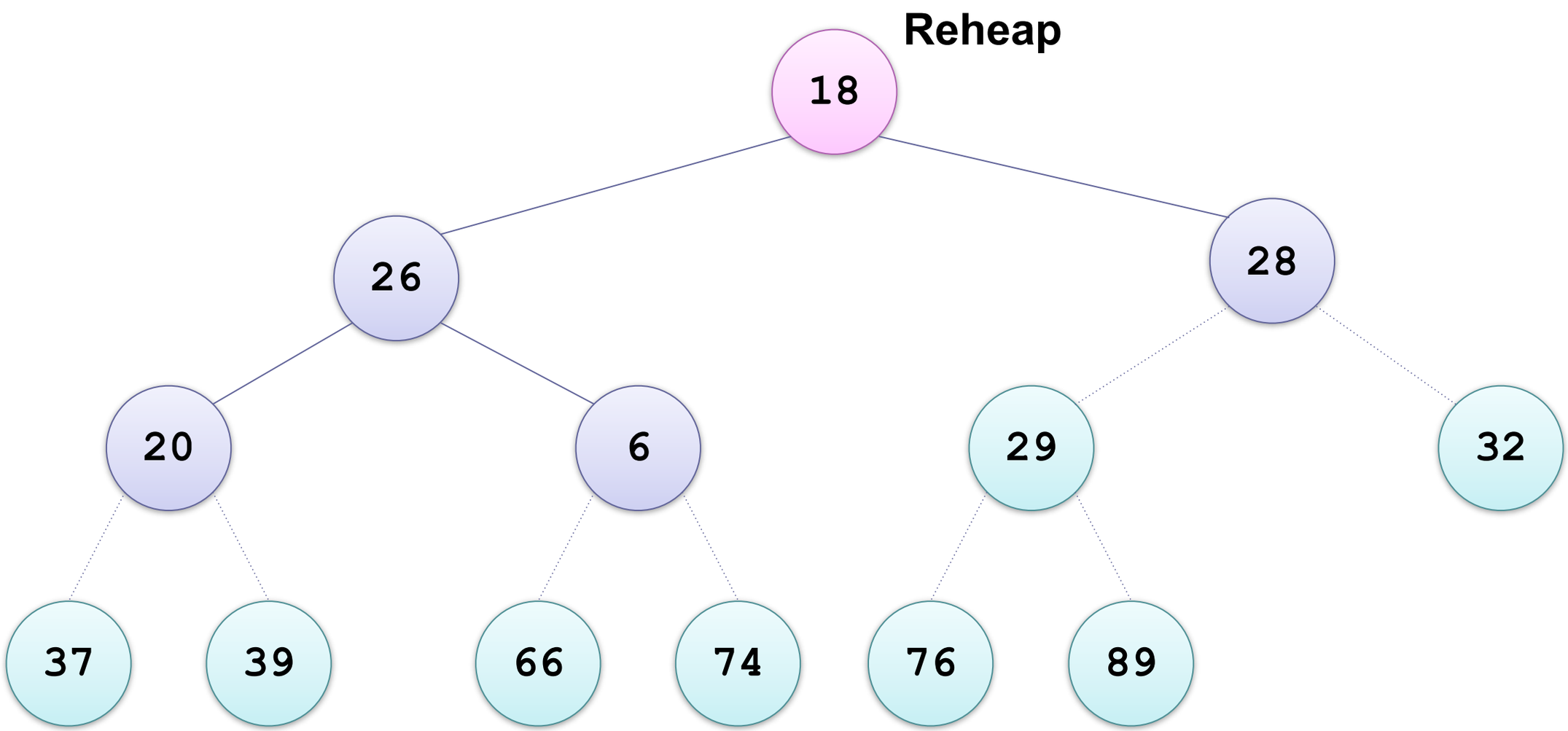
Trace of Heapsort (cont.)



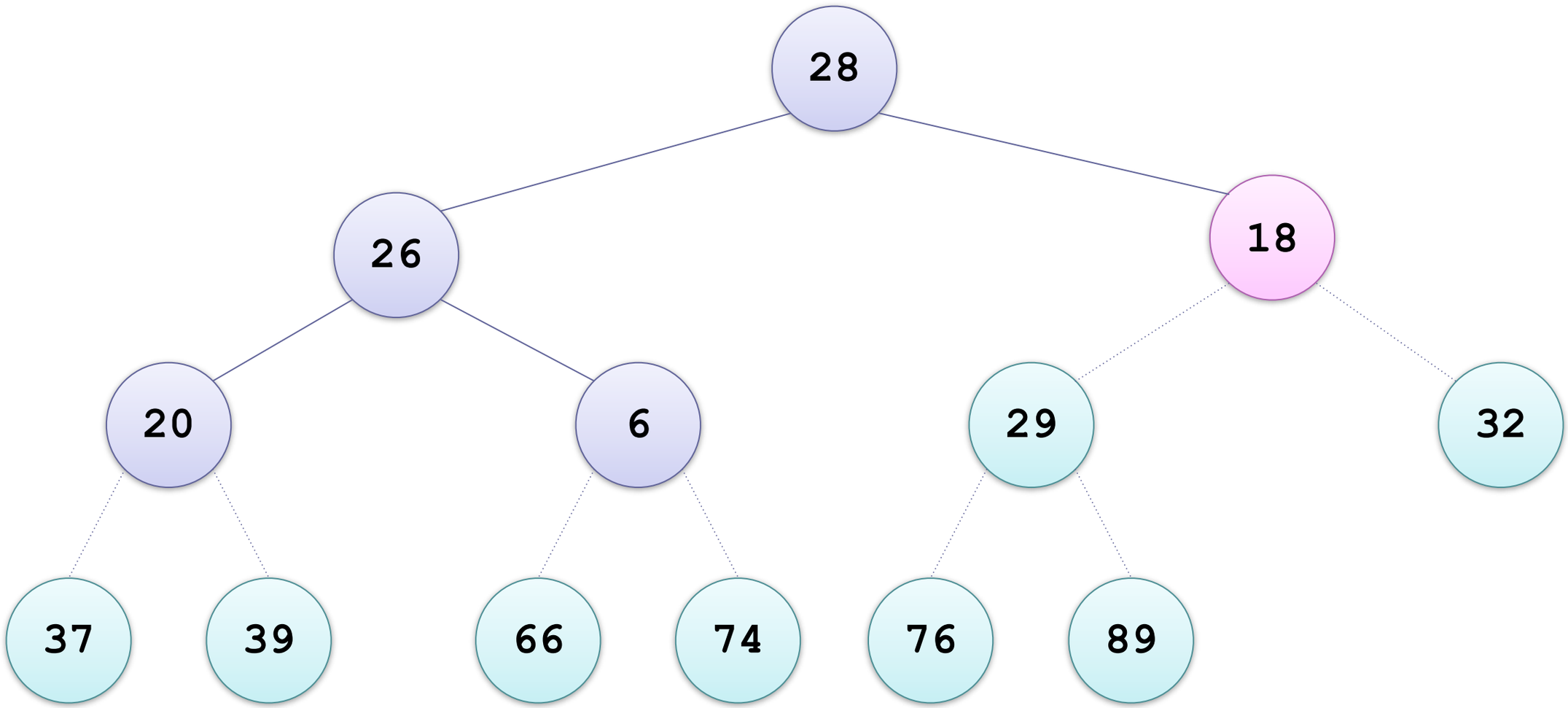
Trace of Heapsort (cont.)



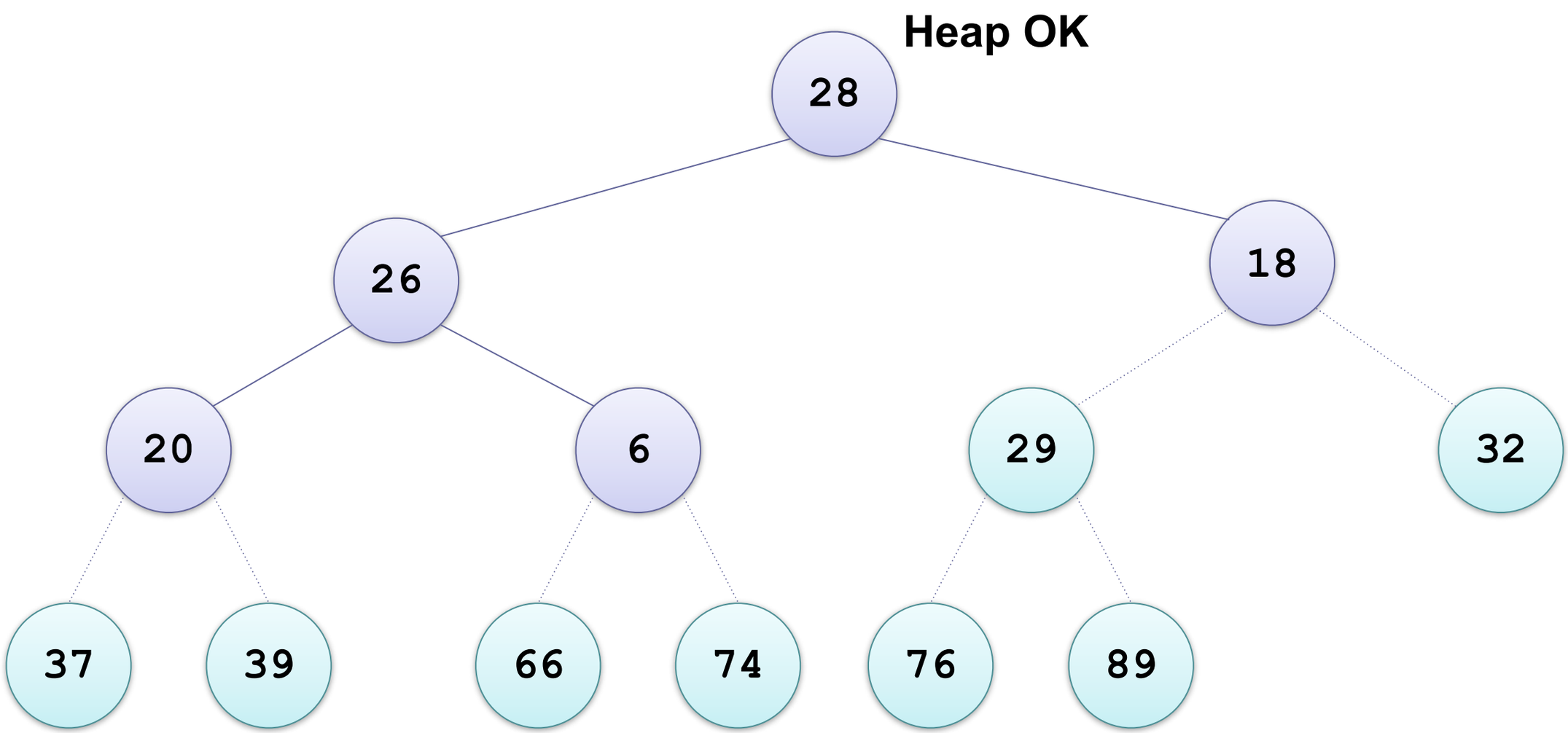
Trace of Heapsort (cont.)



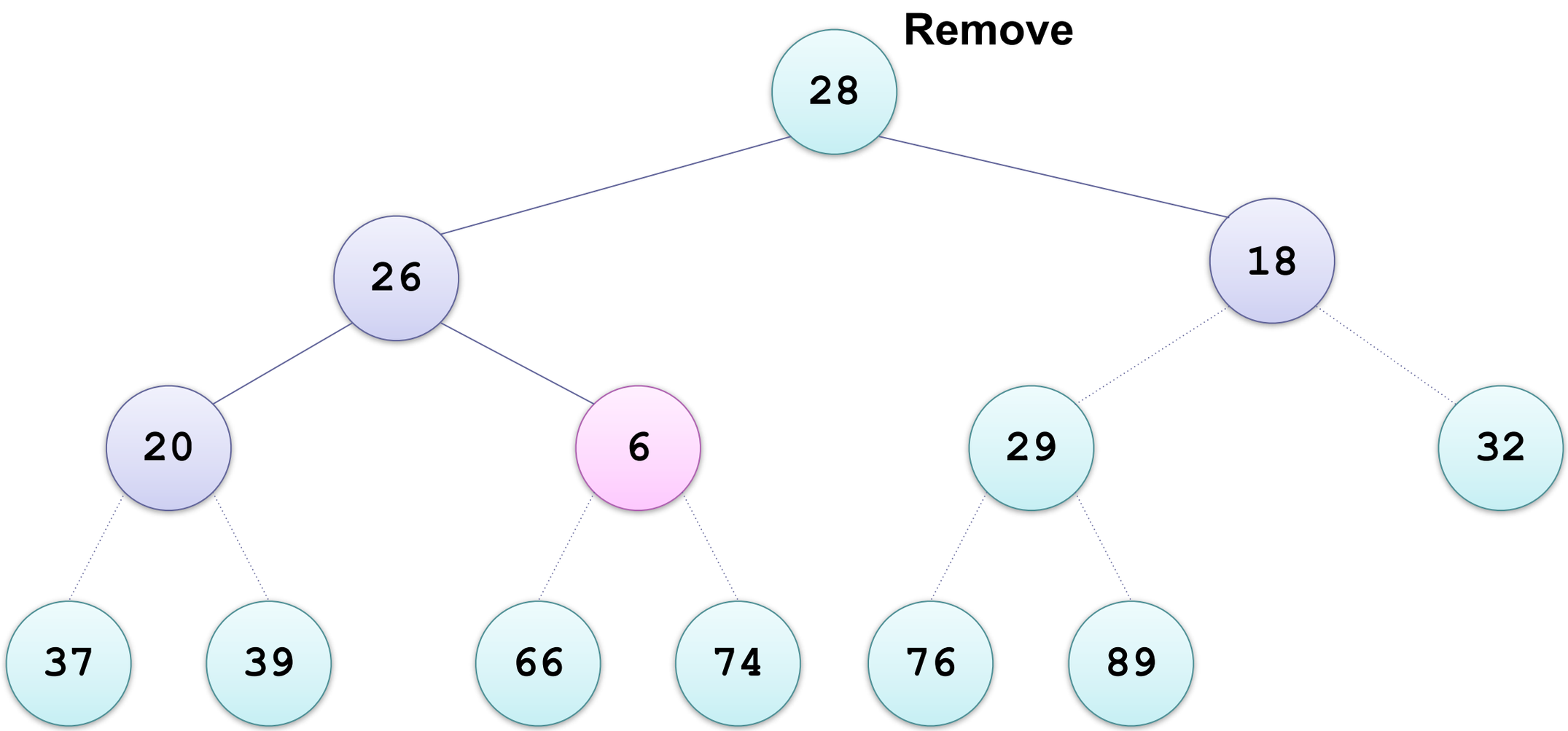
Trace of Heapsort (cont.)



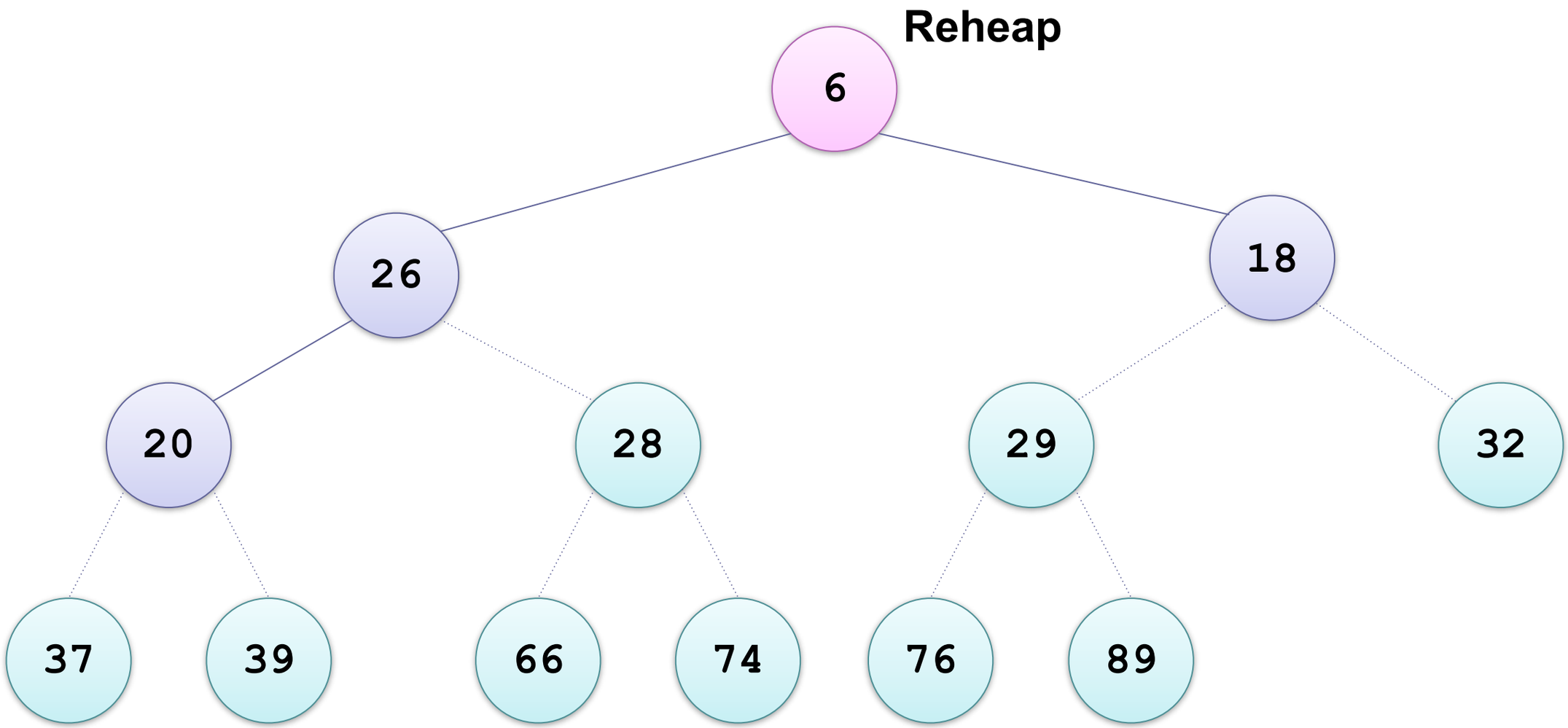
Trace of Heapsort (cont.)



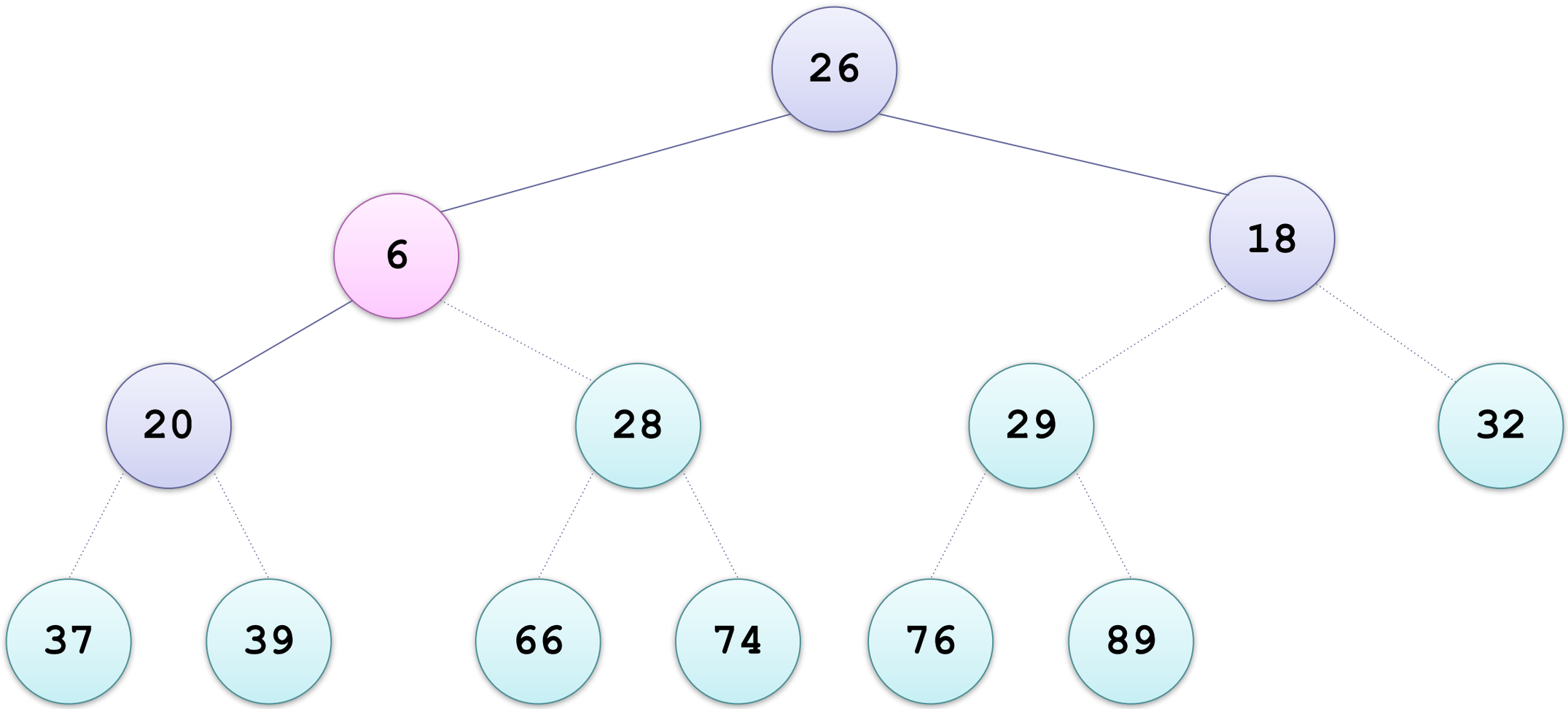
Trace of Heapsort (cont.)



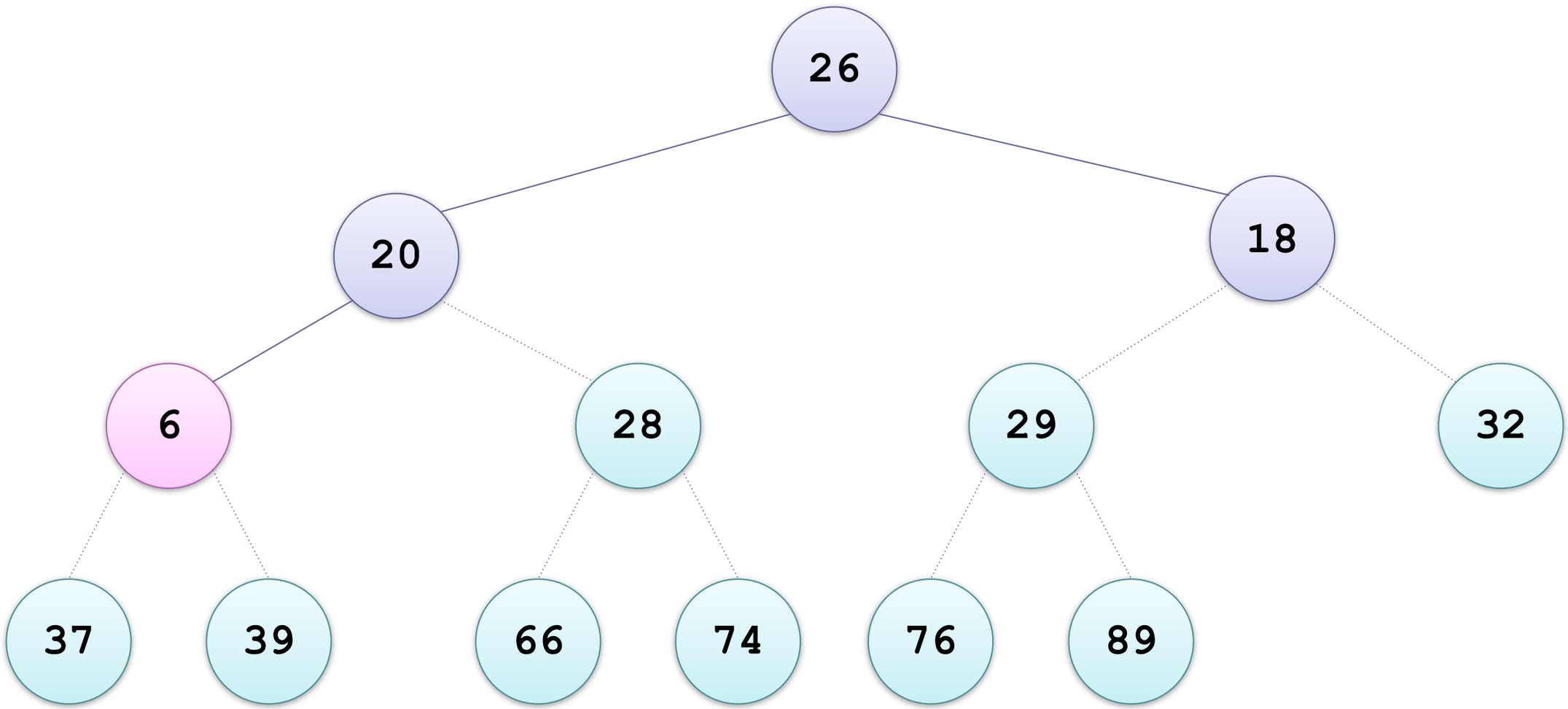
Trace of Heapsort (cont.)



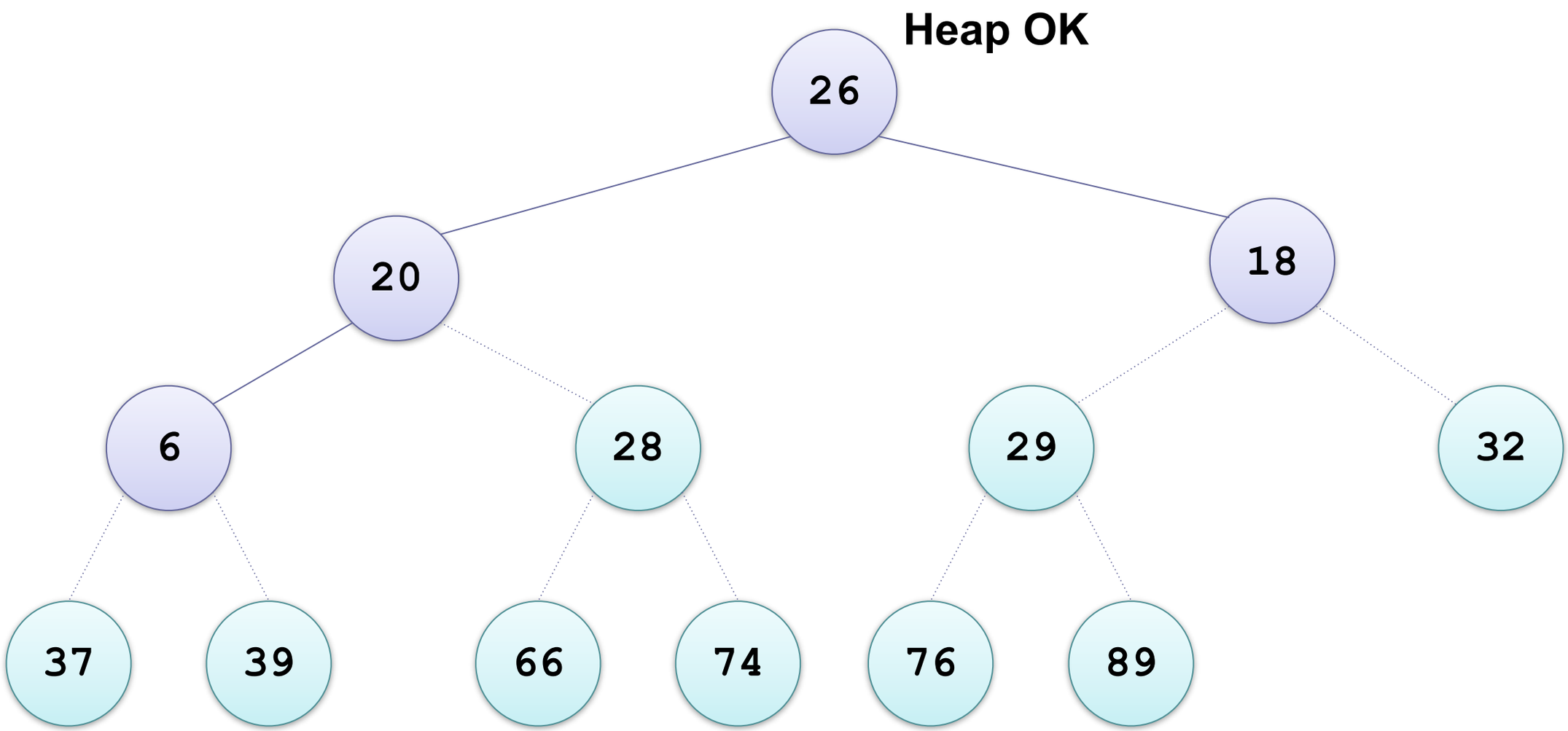
Trace of Heapsort (cont.)



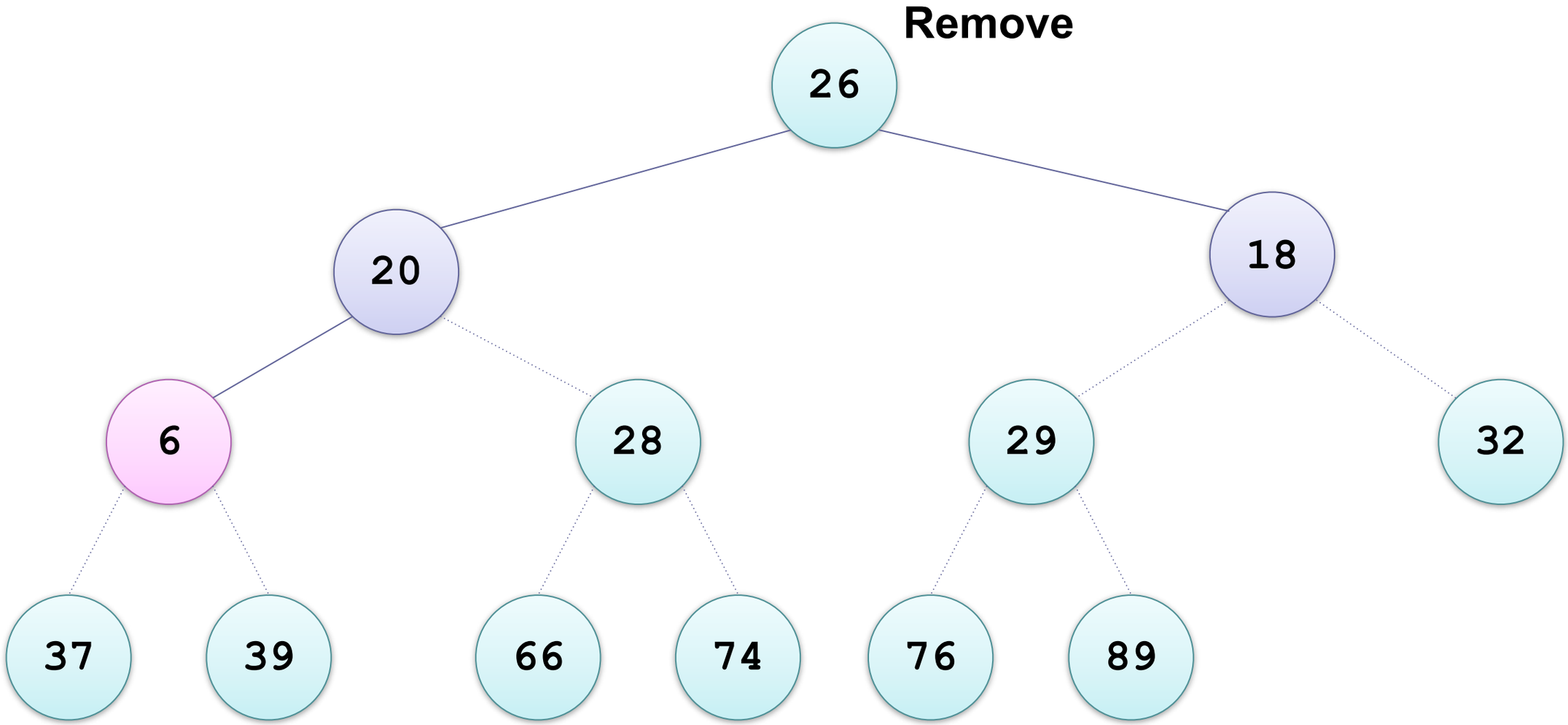
Trace of Heapsort (cont.)



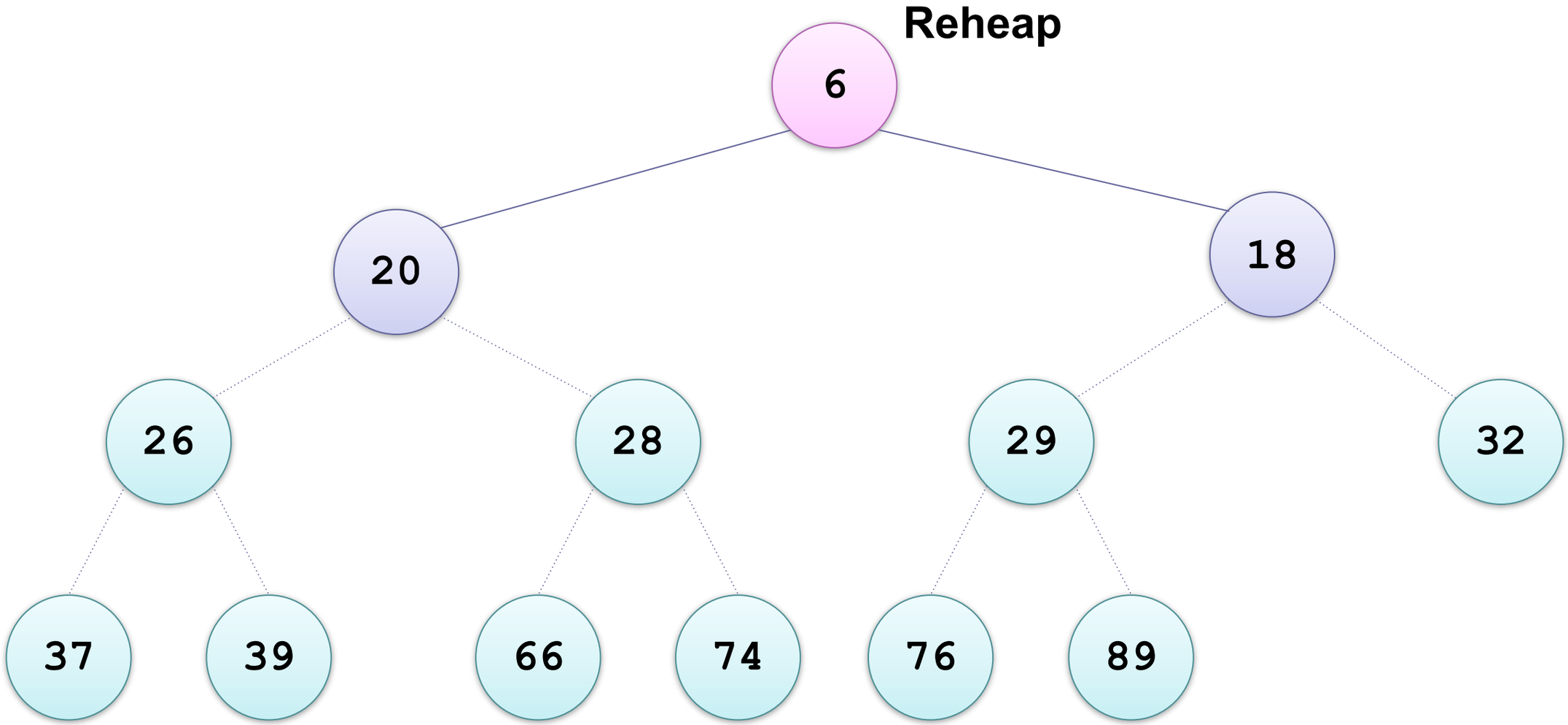
Trace of Heapsort (cont.)



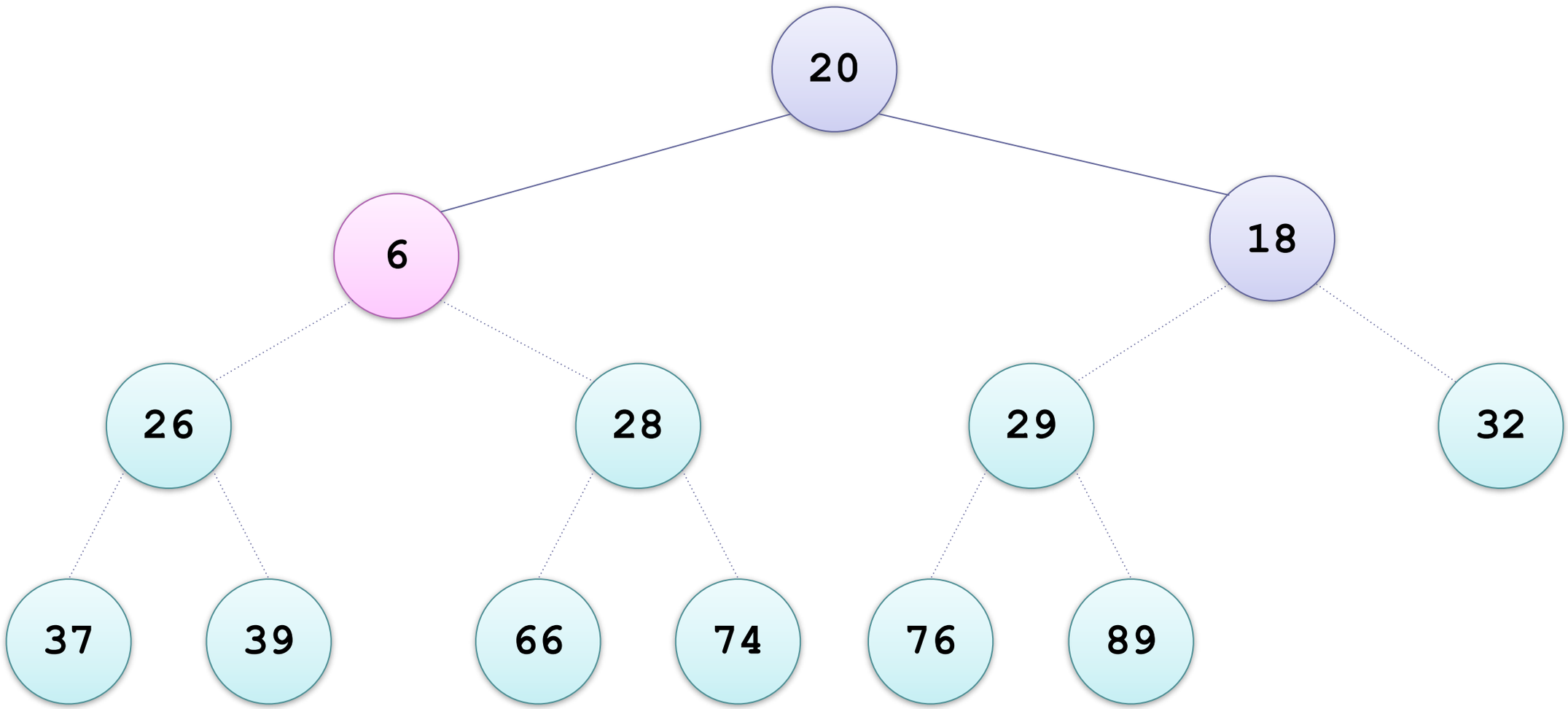
Trace of Heapsort (cont.)



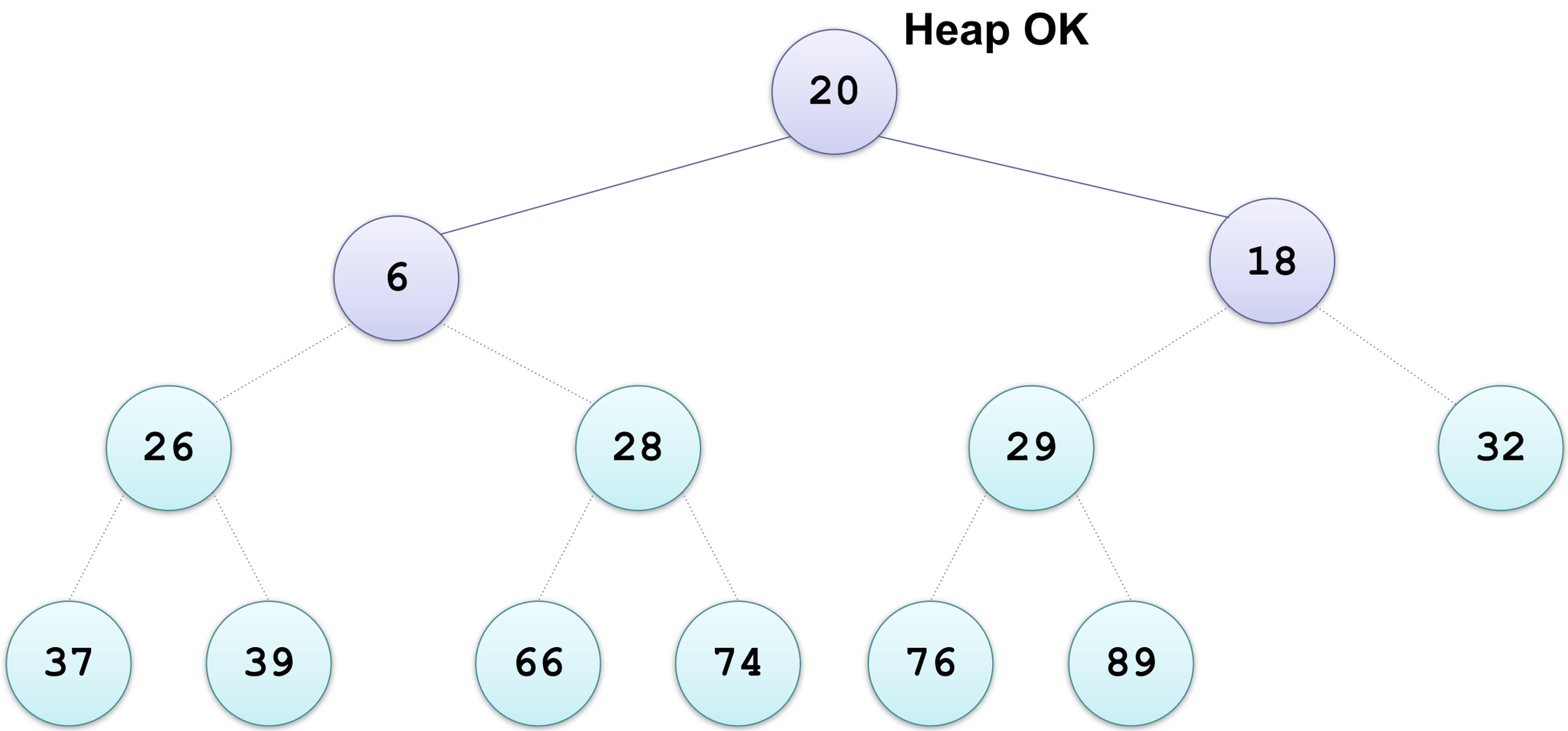
Trace of Heapsort (cont.)



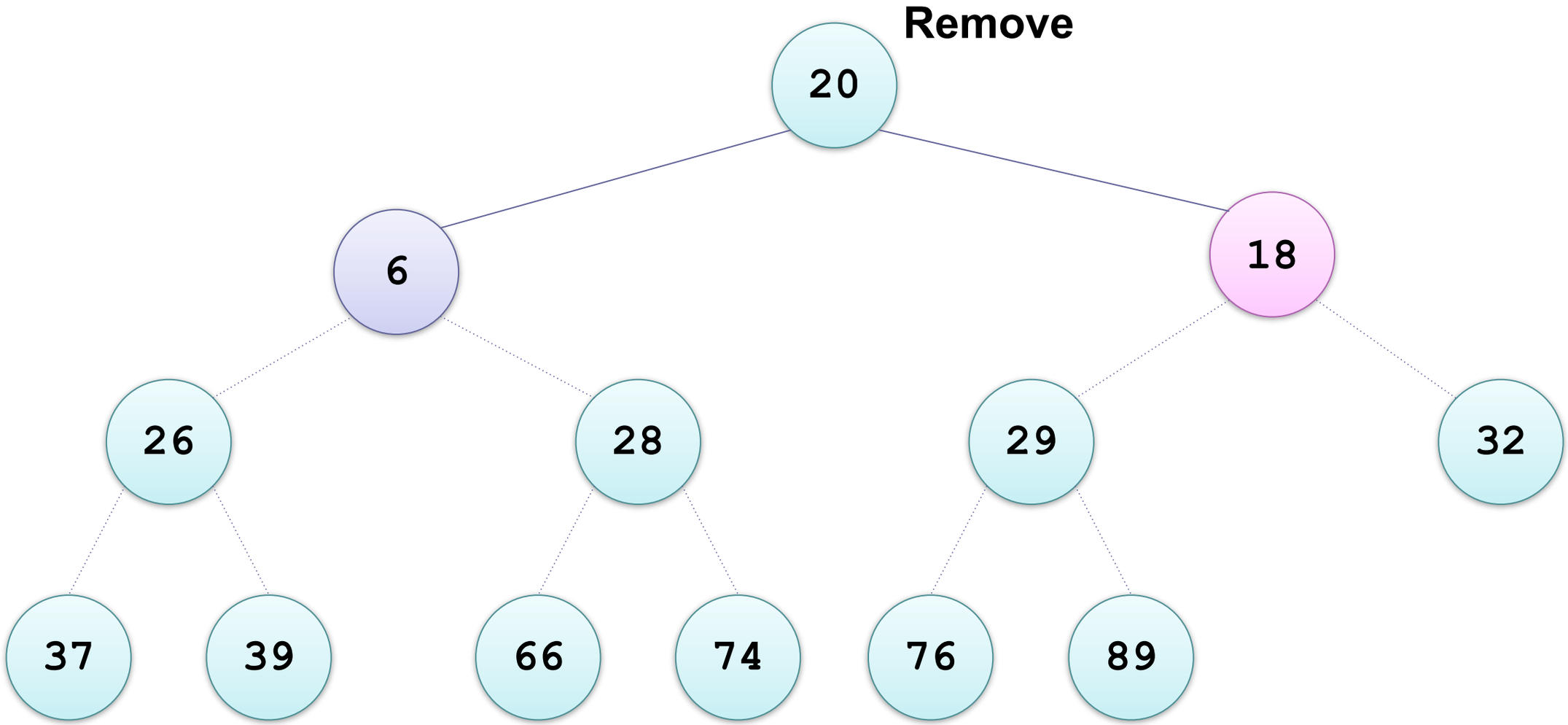
Trace of Heapsort (cont.)



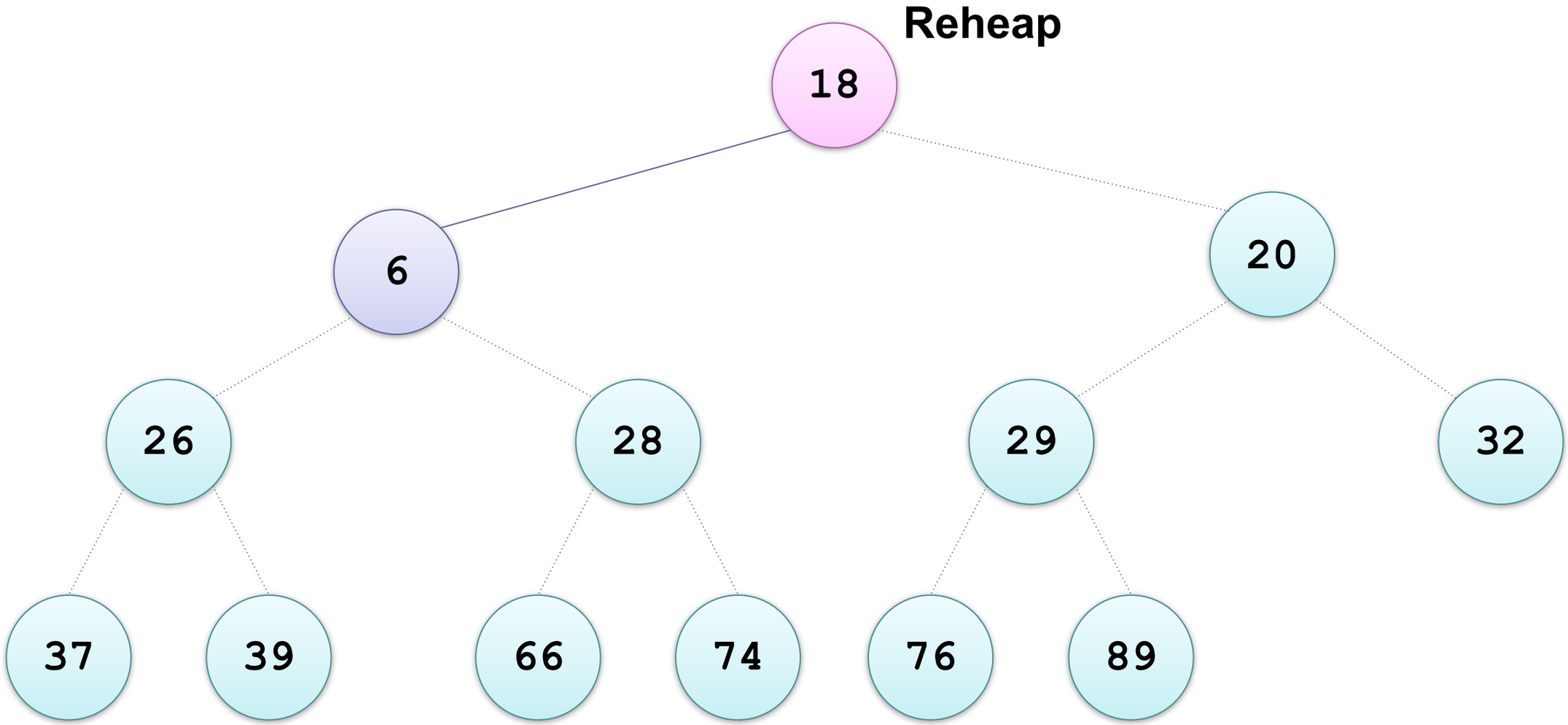
Trace of Heapsort (cont.)



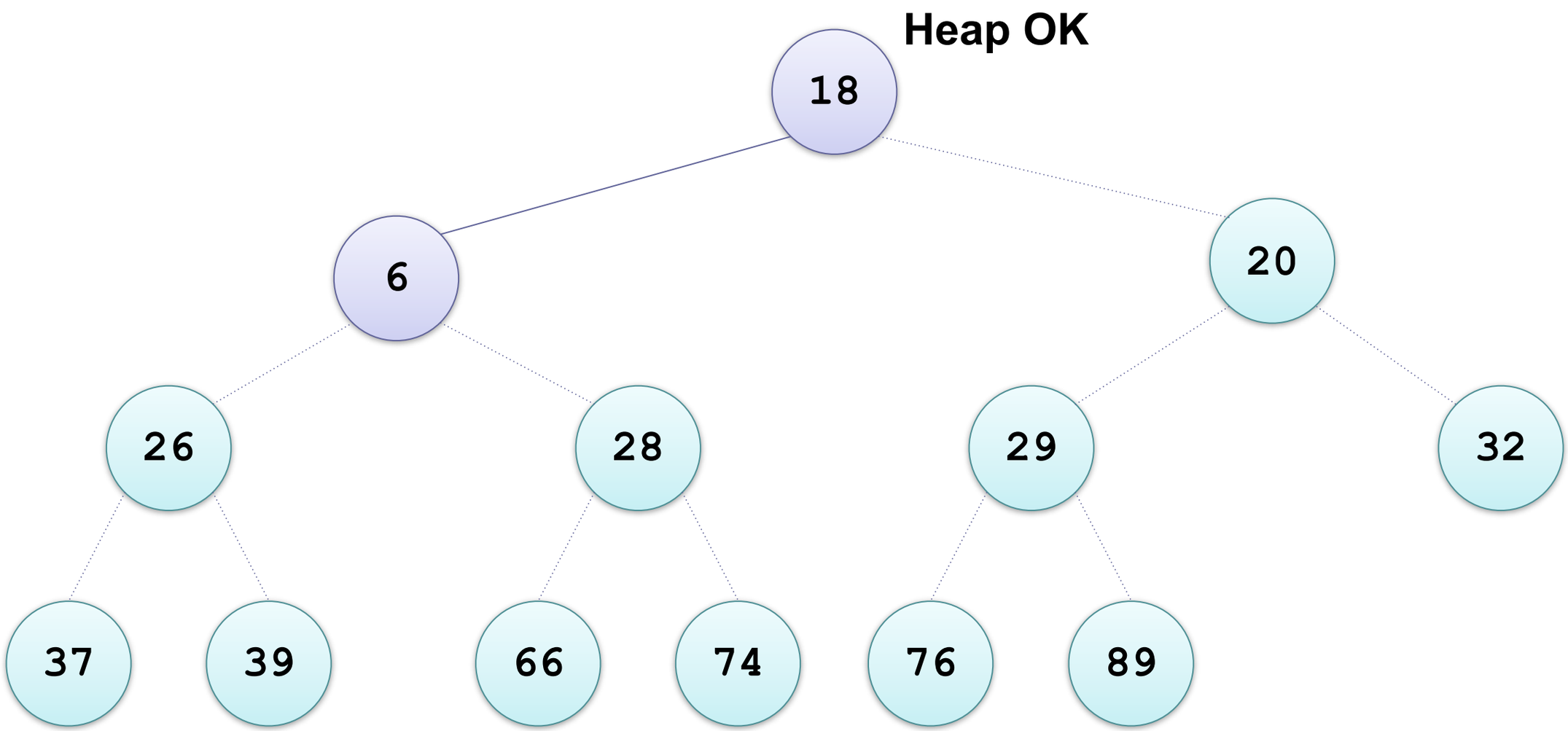
Trace of Heapsort (cont.)



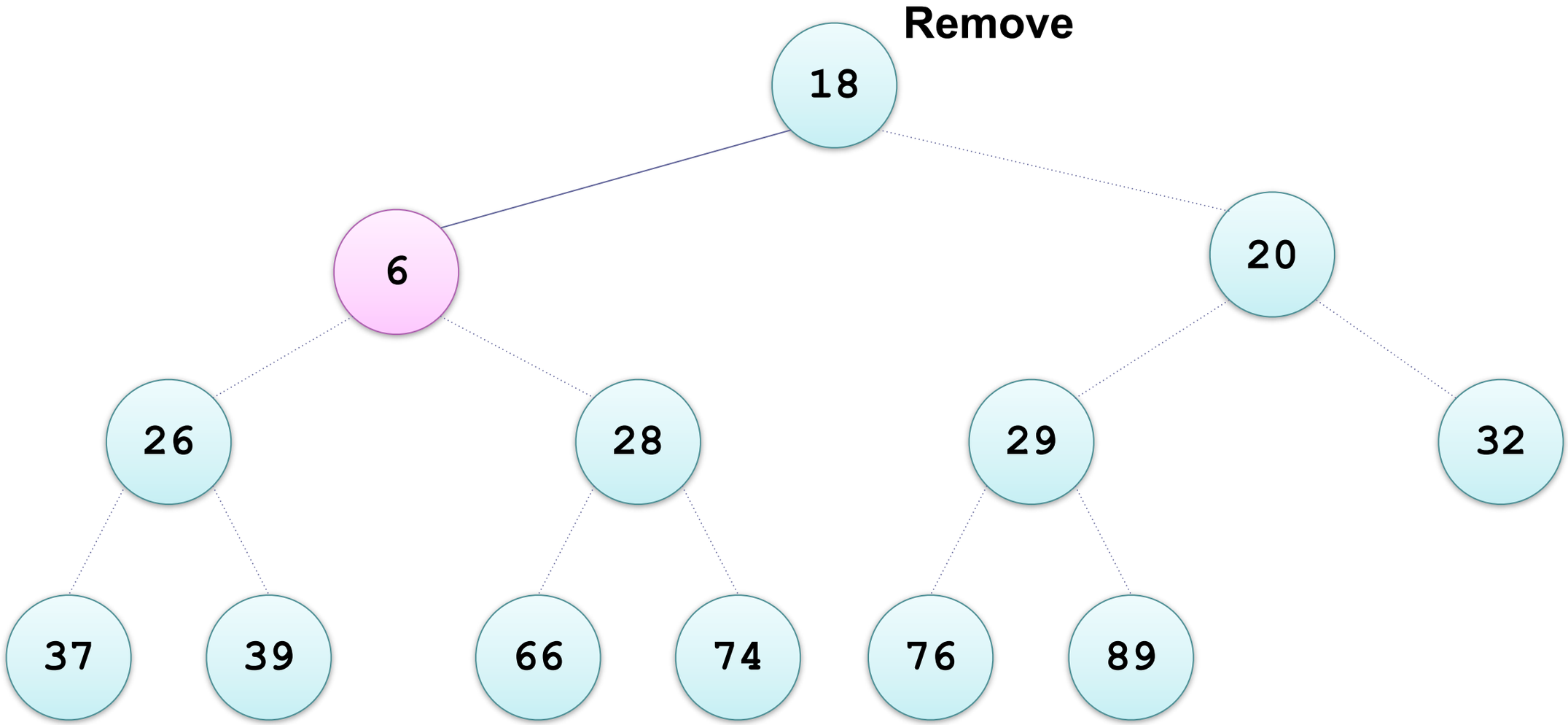
Trace of Heapsort (cont.)



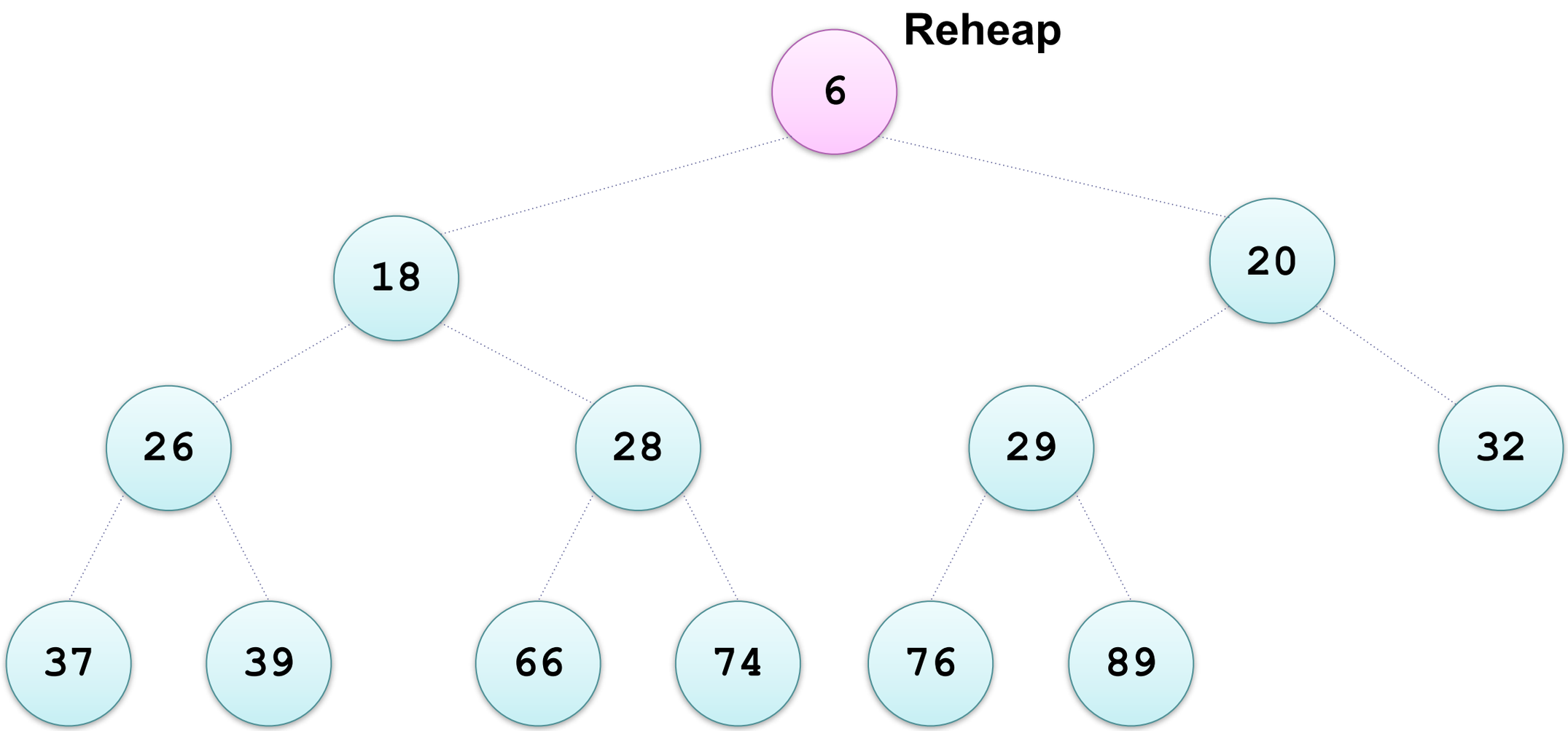
Trace of Heapsort (cont.)



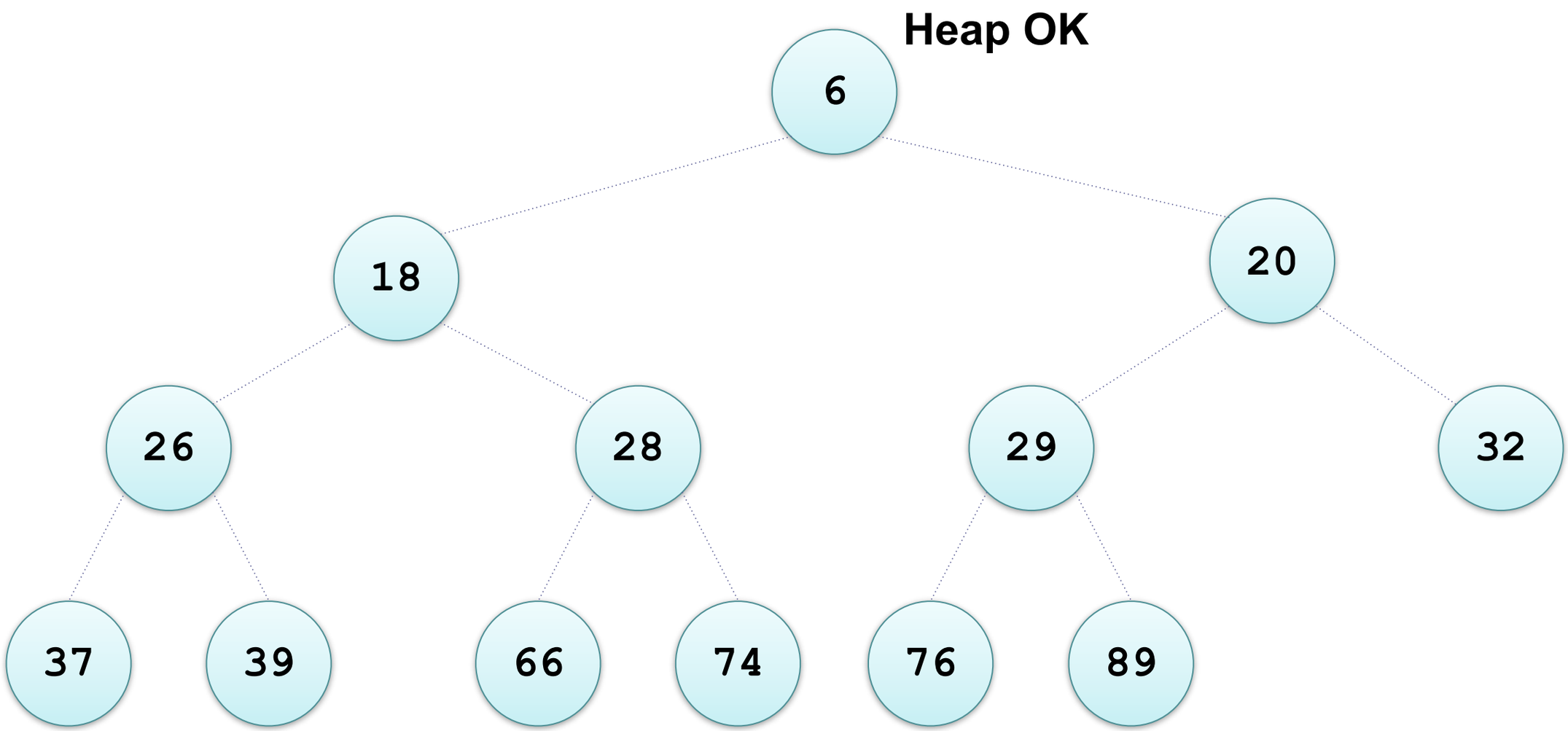
Trace of Heapsort (cont.)



Trace of Heapsort (cont.)



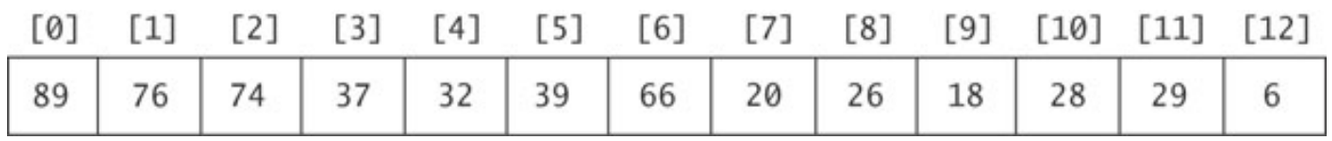
Trace of Heapsort (cont.)

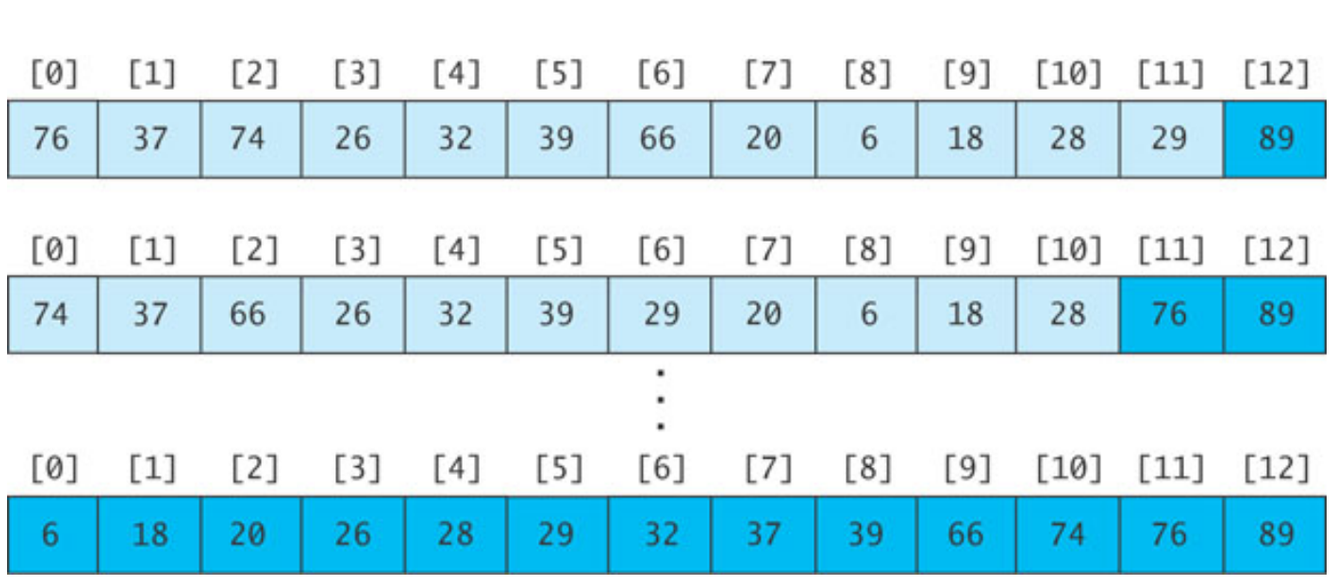


Heapsort, implementering

Vi implementerar heapen som ett fält:

- fältet är uppdelat i en heap-del och en sorterad lista
- varje element vi tar bort flyttas till slutet av fältet
- heap-delen av fältet minskar successivt





Heapsort, algoritm

Algoritm för In-Place Heapsort

1. while n is less than table.length:
2. increment n by 1
   (this inserts a new item into the heap)
3. restore the heap property
4. while the heap is not empty:
5. remove the first item from the heap by
   swapping it with the last item in the heap
6. restore the heap property

Analys av Heapsort

Heapar är logaritmiska:

- stoppa in/ta bort element är O(log n)
- att stoppa in n element är O(n log n)
- att ta bort n element är också O(n log n)
- O(n log n) + O(n log n) = O(n log n)
- algoritmen kräver inget extra minne