Hamming's Problem:

Given: a list of prime numbers.
Output: a list of numbers in increasing order such that a number is output if and only if its prime factors are contained in the given list of prime numbers.

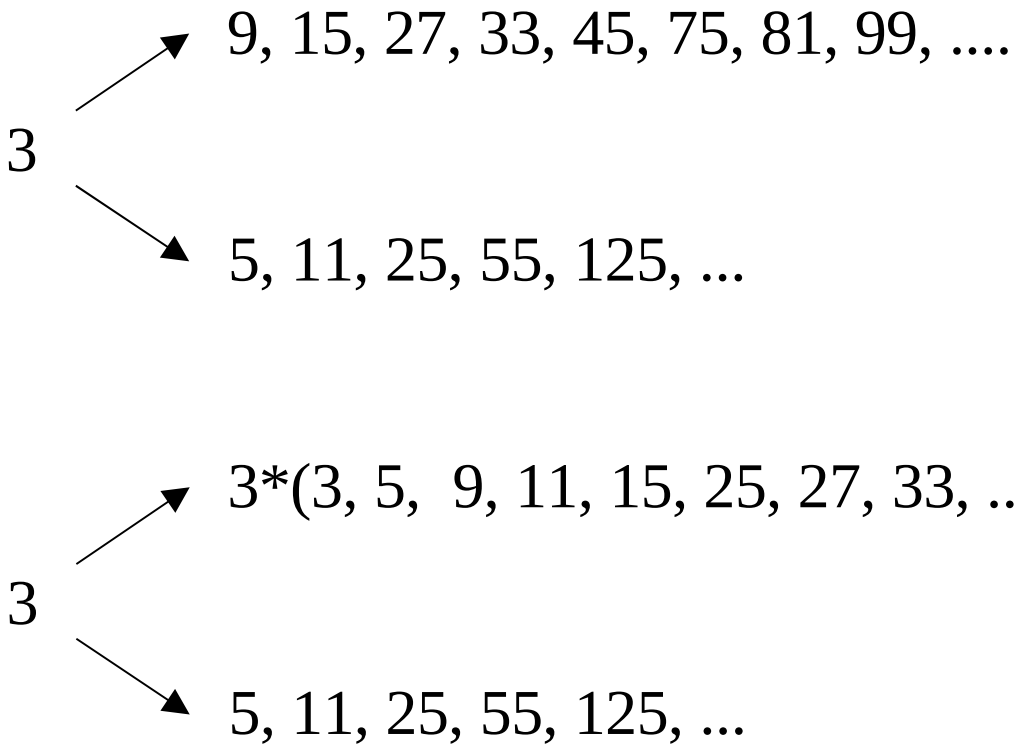
Example: input is 3, 5, 11

output is 3, 5, 9, 11, 15, 25, 27, 33, 45, 55, 75, 81, 99, 121, ...

What to do?

Consider all numbers from 1,2,3... and check for prime factors

Too Slow!!!




```
int *hamming (int *primes) {
    if (primes == NULL) return NULL;
    return concat(first(primes), merge(first(primes)*hamming(primes),
                                       hamming(rest(primes))));
}
```

first(primes) = first number in list 'primes'
rest(primes) = list 'primes' with the first number removed
merge(list-1,list-2) = increasing list of numbers taken from increasing lists 'list-1' and 'list-2'
concat(n, list-1) = add the token 'n' to the list 'list-1'

Bad News!

The call `hamming(primes)` requires calling `hamming(primes)`

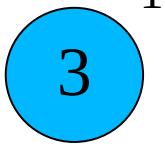
So no output is ever seen!

What to do?

Suspend the calls to `hamming(primes)` using a priority list!

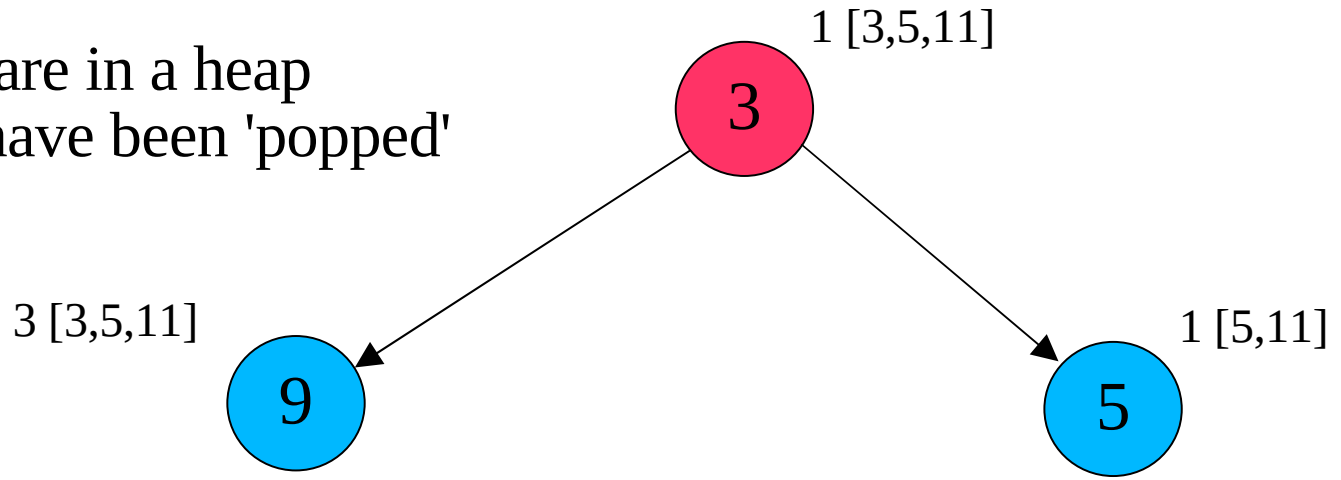
We can implement a priority list using a heap.

Blue nodes are in a heap

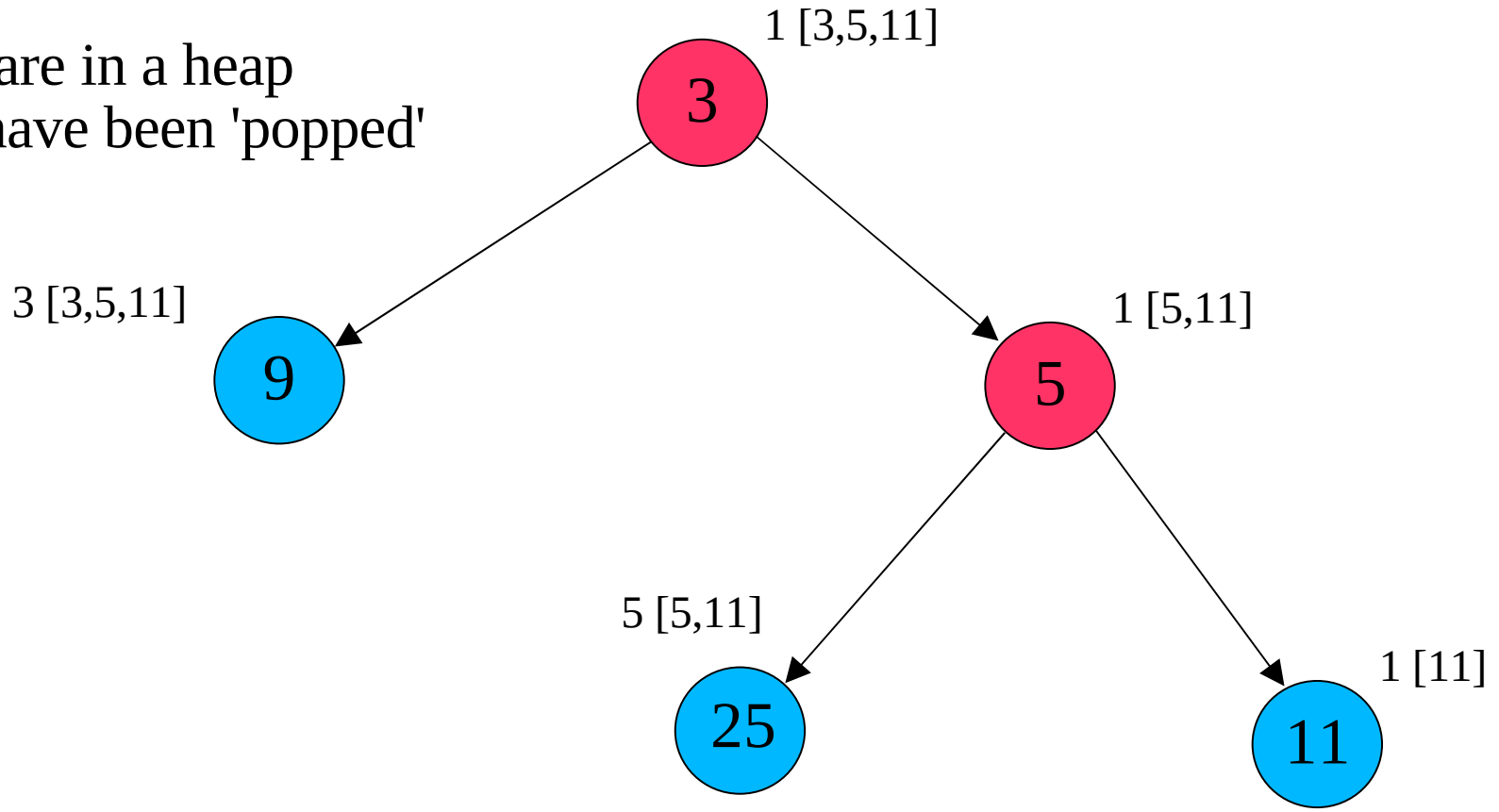


1 [3,5,11]

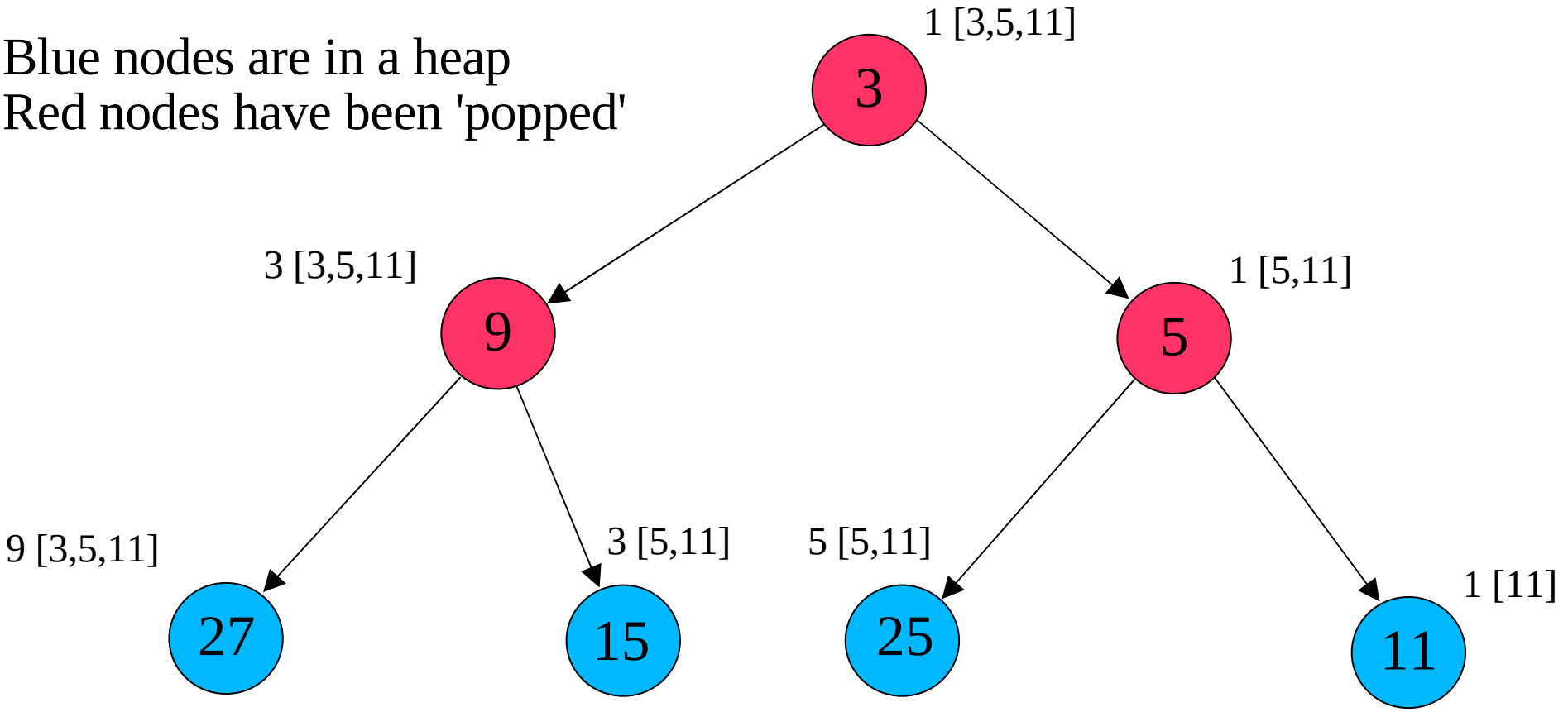
Blue nodes are in a heap
Red nodes have been 'popped'

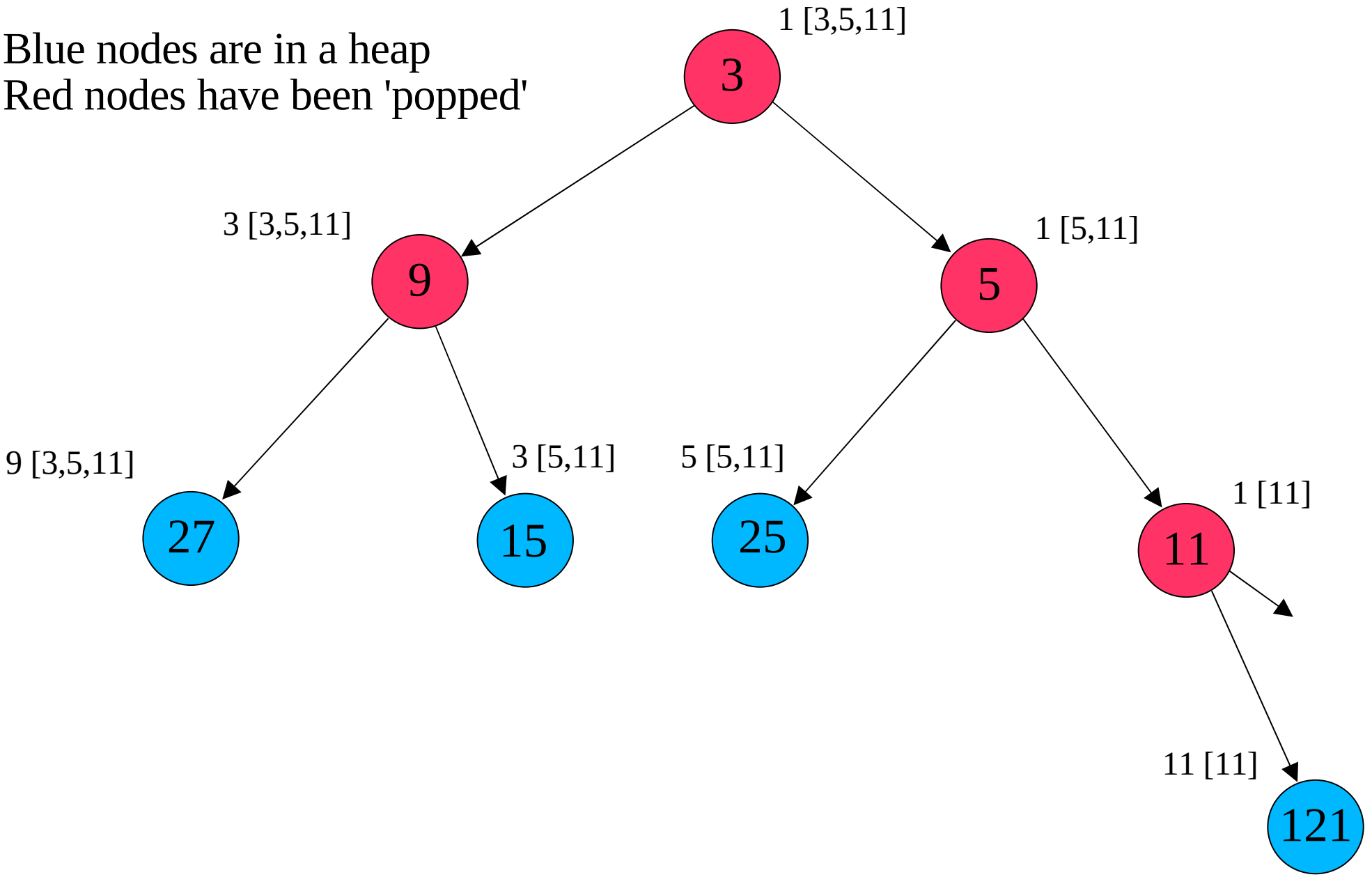


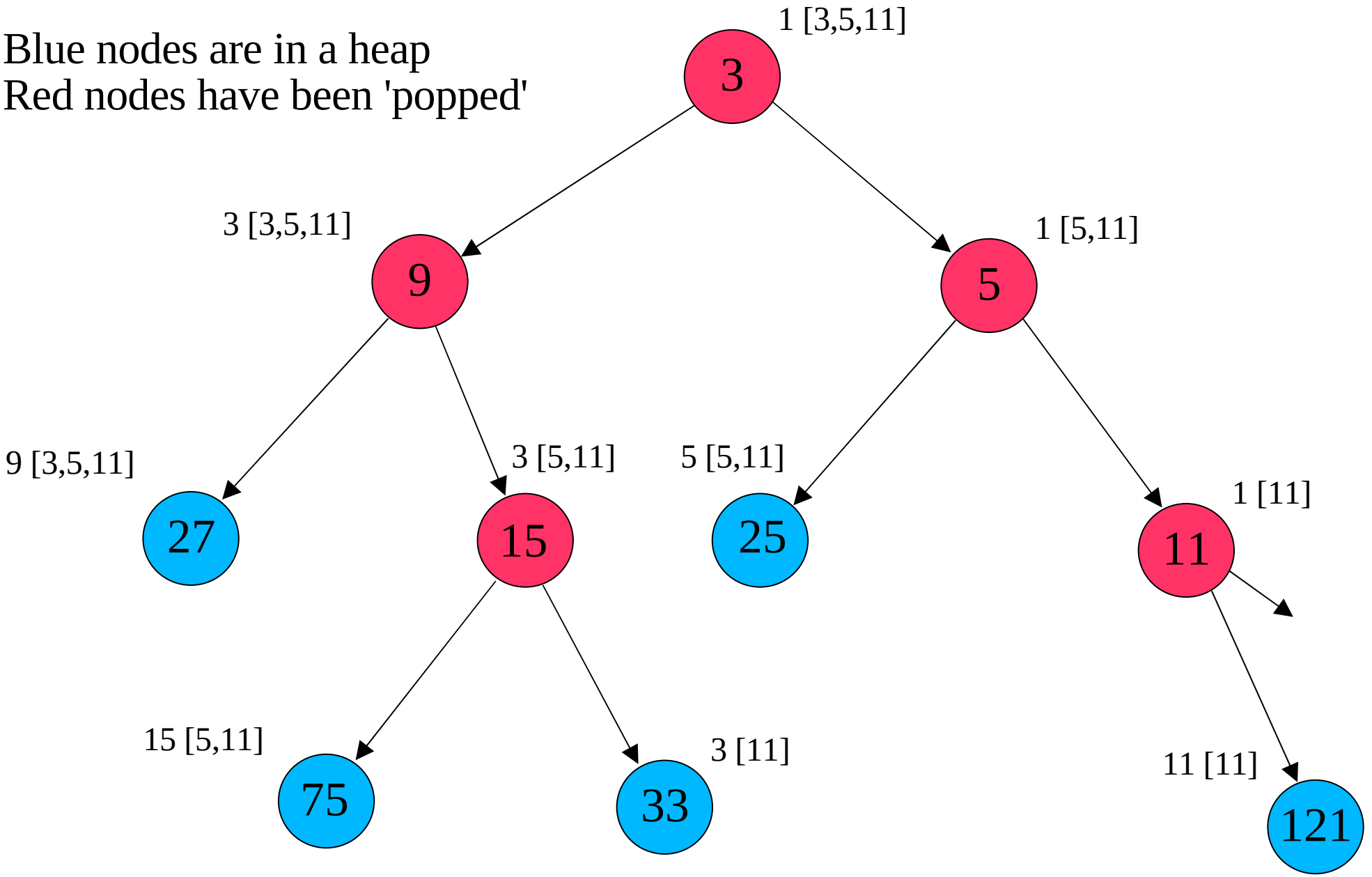
Blue nodes are in a heap
Red nodes have been 'popped'



Blue nodes are in a heap
Red nodes have been 'popped'




Blue nodes are in a heap
Red nodes have been 'popped'
1 [3,5,11]
3
3 [3,5,11]
9
1 [5,11]
5
9 [3,5,11]
27
3 [5,11]
15
5 [5,11]
25
1 [11]
11
11 [11]
121


Blue nodes are in a heap
Red nodes have been 'popped'
1 [3,5,11]
3
3 [3,5,11]
9
1 [5,11]
5
9 [3,5,11]
27
3 [5,11]
15
5 [5,11]
25
1 [11]
11
15 [5,11]
75
3 [11]
33
11 [11]
121