**Part 1:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 27 | 10 | 12 | 20 | 25 | 13 | 15 | 22 |
| 10 | 27 | 12 | 20 | 13 | 25 | 15 | 22 |
| 10 | 12 | 20 | 27 | 13 | 15 | 22 | 25 |
| 10 | 12 | 13 | 15 | 20 | 22 | 25 | 27 |

**Part 2:**

Theta(nlogn)

**Part 3:**

4 2 5 9 8 10 17 20 25 30 15

**Part 4:**

2n-1 comparisons

The worst case occurs when the loop is exited, because one of the indices— say, i—has reached its exit point h + 1 whereas the other index j has reached m, 1 less than its exit point. For example, this can occur when the first m − 1 items in V are placed first in S, followed by all h items in U, at which time the loop is excted because i equals h + 1.

The basic operation is the comparison that takes place in merge. Because the number of comparisons increases with h and m, and h and m increase with n, we have the following:

Basic operation: the comparison that takes place in merge.

Input size: n, the number of items in the array S.

The total number of comparisons is the sum of the number of comparisons in the recursive call to mergesort with U as the input, the number of comparisons in the recursive call to mergesort with V as the input, and the number of comparisons in the top-level call to merge.