1. We can see that the smallest element appears \(2n - 1\) times in \(C\) \(= \min\{k_1, k_1\}, \min\{k_1, k_2\}, \ldots, \min\{k_1, k_n\}, \min\{k_2, k_1\}, \min\{k_3, k_1\}, \ldots, \min\{k_n, k_1\}\). Similarly, the second smallest element appears \(2n - 3\) times in \(C\) and the \(i\)th smallest element appears \(2n - 2i + 1\) times in \(C\). Now the median of \(C\) is the \(j\)th smallest element such that \(\sum_{i=1}^{j} 2n - 2i + 1 = \frac{n^2}{2}\).

Now, the median of \(C\) can be obtained by finding the \(j\)th smallest element in \(S\).

**Complexity = \(O(n)\).**

2. Sort the sets \(A\) and \(B\) using the radix sort algorithm. These sorts can be done in \(O(n)\) time. Now apply the algorithm of Problem 3 to compute \(A \cap B\) in \(O(n)\) time. If \(A \cap B = \emptyset\) then the sets are disjoint. Total time taken is \(O(n) + O(n) = O(n)\).

3. Let \((p_1, p_2) = (2, 1), (w_1, w_2) = (x, 1), m = 1, x > 1\). Then

\[
\frac{F^*(I)}{F(I)} = \frac{1}{2^1} = \frac{x}{2} \to \infty \text{ when } x \to \infty
\]

4. One possible minimum spanning tree has the following edges: \((1, 3), (1, 4), (2, 3),\) and \((3, 5)\). The total weight is 14.


In stage 1, node 1 has the minimum \(dist\) value and hence is inserted into the set \(S\). Nodes 3 and 4 are neighbors of 1 and hence we have to check if the \(dist\) values of these nodes have to be modified. Since \(dist[3] > dist[1] + W(1,3)\), we change \(dist[3]\) to 8. Likewise we set \(dist[4] = 5\).

In stage 2, node 4 has the minimum \(dist\) value and hence is inserted into \(S\). Nodes 2, 5, and 3 are neighbors of 4. The new \(dist\) values of these nodes become: \(dist[2] = 10; dist[3] = 6; dist[5] = 10\).

In stage 3, node 3 has the minimum \(dist\) value and it becomes a part of \(S\). All the neighbors of 3 are already in \(S\) and the \(dist\) values of the nodes 2 and 5 do not change.

In stage 4, we have two nodes both having the same \(dist\) value. We could pick one arbitrarily and insert it into \(S\). Let 2 be this node. The \(dist\) value of 5 does not change.

In stage 5, the node 5 also enters \(S\). Algorithm terminates then.

Thus the shortest paths from \(s\) to the nodes 1, 2, 3, 4, and 5 are 2, 10, 6, 5, and 10, respectively.