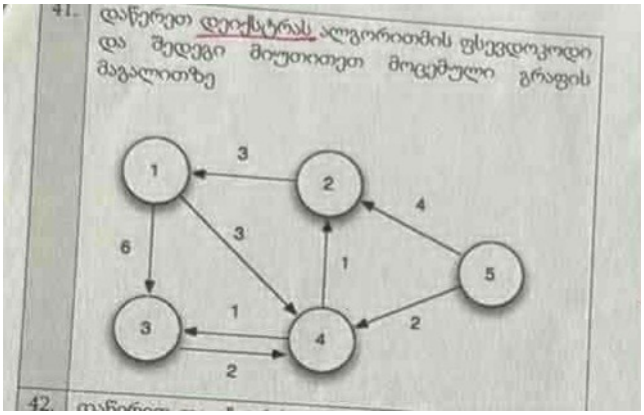
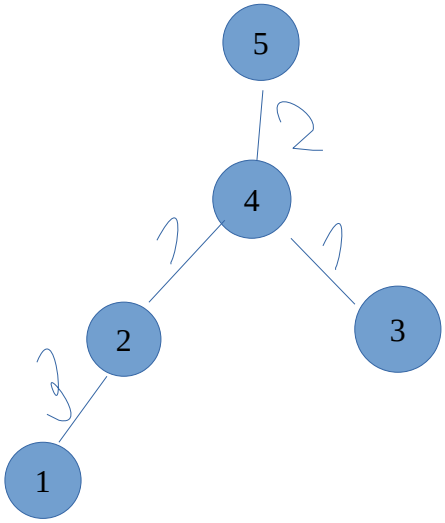
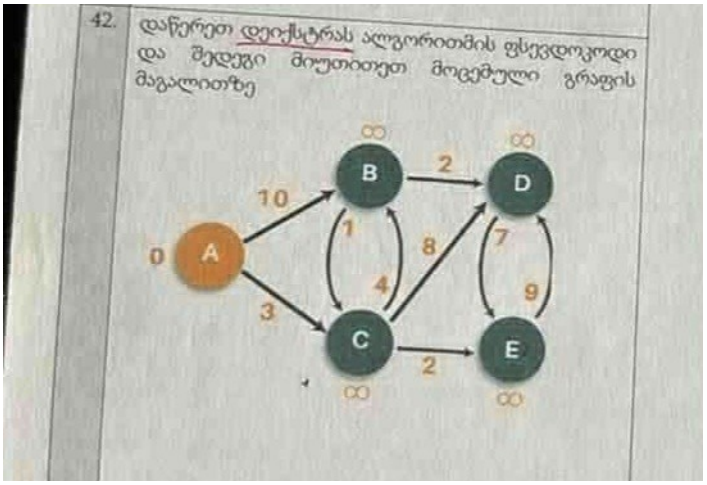



```
void deixtra(vector<ipair> adj[], int V, int s) {
    vector<bool> visit(V, false);
    vector<int> dist(V, 999);
    vector<int> parent(V, -1);

    vector<ipair>::iterator it;
    priority_queue<ipair, vector<ipair>, greater<ipair>> pq;

    dist[0] = 0;
    pq.push(make_pair(dist[0], 0));

    while (!pq.empty()) {
        int u = pq.top().second;
        pq.pop();

        visit[u] = true;

        for (it = adj[u].begin(); it != adj[u].end(); ++it) {
            int v = (*it).first;
            int wt = (*it).second;


            if (visit[v] == false && dist[v] > dist[u] + wt) {
                parent[v] = u;
                dist[v] = dist[u] + wt;
                pq.push(make_pair(dist[v], v));
            }
        }
    }

    for (int i = 1; i < V; ++i)
        cout << parent[i] << "->" << i << " " << dist[i] << endl;
}
```

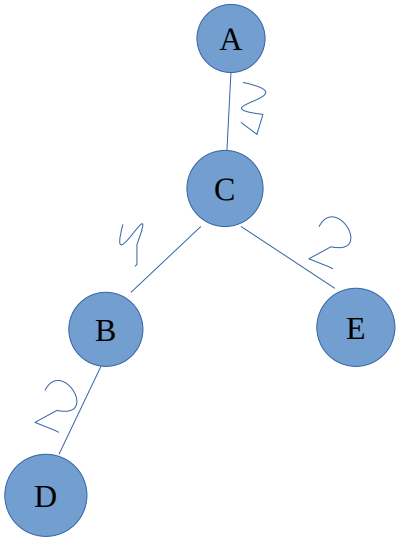
იტერაცია		D[1]	D[2]	D[3]	D[4]
	5		4		2
	5,4		3	3	
	5,4,2	6		3	
	5,4,2,3	6			


5
4
2
3
1

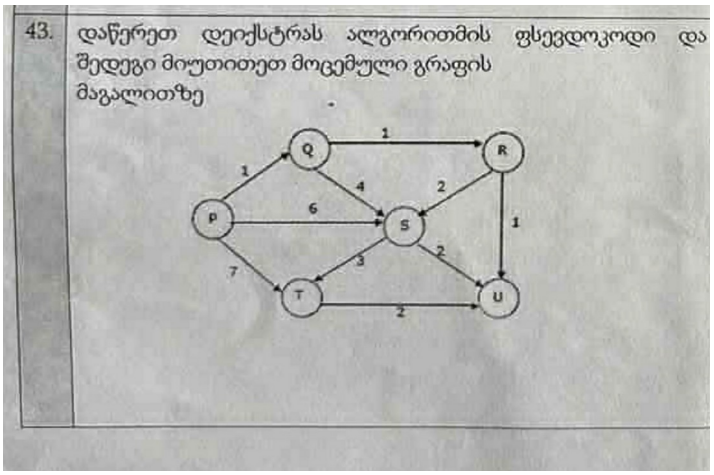
42.



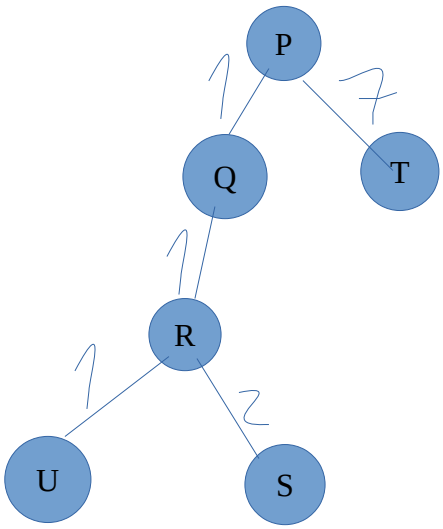
იტერაცია		დისტანცია[B]	დისტანცია[D]	დისტანცია[C]	დისტანცია[E]
	A	10		3	
	A,C	7	11		5
	A,C,E	7	11		
	A,C,E,B		9		



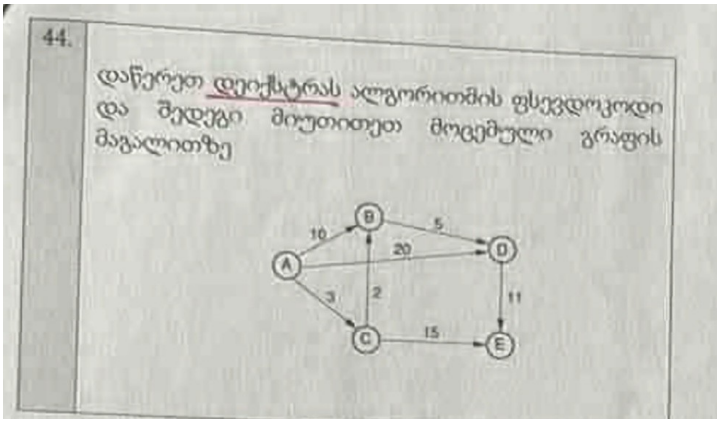
43.



იტერაცია		დისტანცია[Q]	დისტანცია[R]	დისტანცია[S]	დისტანცია[T]	დისტანცია[U]
	P	1		6	7	
	P,Q		2	5	7	
	P,Q,R			4	7	3
				4	7	



44.



იტერაცია		დისტანცია[B]	დისტანცია[C]	დისტანცია[D]	დისტანცია[E]
	A	10	3	20	
	A,C	5		20	18
	A,C,B			10	18