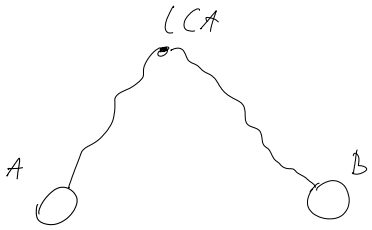
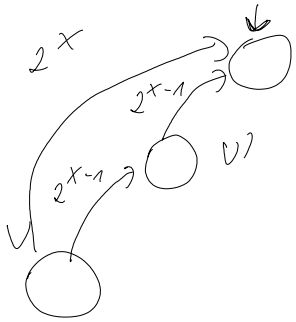
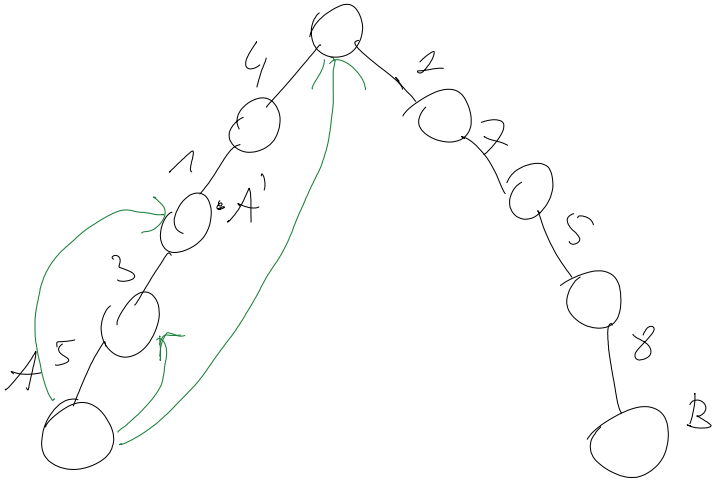
```
int n, skip, Q;
vector<PII> G[MAX_N];
int par[MAX_N], depth[MAX_N], val[MAX_N];
int ancestor[MAX_N][LOG];
int mini[MAX_N][LOG];
```

```
int goUp(int v, int dist) {
    int res = inf;
    for (int k = LOG-1; k >= 0; k--) {
        if (dist & (1 << k)) {
            res = min(res, mini[v][k]);
            v = ancestor[v][k];
        }
    }
    return res;
}
```

```
for (int k = 1; k < LOG; k++) {
    for (int x = 1; x <= n; x++) {
        ancestor[x][k] = ancestor[ancestor[x][k-1]][k-1];
        mini[x][k] = min(mini[x][k-1], mini[ancestor[x][k-1]][k-1]);
    }
}

cin >> Q;
for (int qq = 0; qq < Q; qq++) {
    int u, v;
    cin >> u >> v;
    int lca = findLca(u, v);

    int dist = depth[u] - depth[lca];
    int ans = goUp(u, dist);

    //cout << "D: " << lca << " " << dist << " " << ans << '\n';

    dist = depth[v] - depth[lca];
    ans = min(ans, goUp(v, dist));
```




$mini[A][0] =$

$mini[A][1] = \min(5,3) = 3$

$mini[A][2] = \min(\min(5,3), \min(1,4)) = 1$

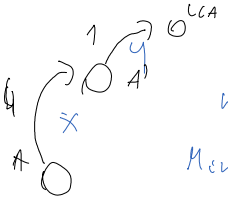
---

$Dist = 5 = 4 + 1$

$5 = 101_{(2)}$

$\downarrow \quad \downarrow \quad \downarrow$

$2^2 \quad 2^1 \quad 2^0$




$\min(x,y)$

Minimum po wszystkich skokach

$dist \;\&\; (1 << k)$

$(1 << k) \equiv 2^k$