Union-find algorithms
Sample code

Bruce Merry

March 2, 2006

```
/* Returns a node that is in the same set as x. The same node
 * will be returned for any x in the same set.
 */
int find(int x)
{
    // identify the root
    int root = x;
    while (parent[root] >= 0) root = parent[root];

    // path compression
    while (x != root)
    {
        int old = x;
        x = parent[x];
        parent[old] = root;
    }
    return root;
}

/* Combines the sets that x and y occupy. Returns true on
 * success, or false if x and y are already in the same set.
 */
bool merge(int x, int y)
{
    // identify roots
    x = find(x);
    y = find(y);
    // check for no-op
    if (x == y) return false;
    // make sure that x is the larger set. parent[x] is the
    // negated size of the set containing x.
    if (parent[x] > parent[y]) swap(x, y);
    // update the size of x
    parent[x] += parent[y];
    // place y under x
    parent[y] = x;
    return true;
}
```