Disjoint Set Union/Find

Ray Seyfarth
School of Computing
University of Southern Mississippi

January 31, 2006
Outline

Motivation

Linked List Representation

Disjoint Set Forests

Heuristics

C Code using Arrays

Example
Disjoint Set Operations

- Make-Set($x$): create a set with only $x$
- Union($x, y$): unite the sets containing $x$ and $y$
- Find-Set($x$): return the representative of the set containing $x$

Initially each item of the universe is placed in a set by itself. A sequence of Union and Find-Set operations develops a collection of disjoint sets.
Connected-Components($G$)

1. for each vertex $v \in V[G]$
2. Make-Set($v$)
3. end-for
4. for each edge $(u, v) \in E[G]$
5. if Find-Set($u$) $\neq$ Find-Set($v$) then Union($u, v$)
6. end-for
Outline

Motivation

Linked List Representation

Disjoint Set Forests

Heuristics

C Code using Arrays

Example
Linked List Representation of Disjoint Sets

- Each set has its first node as its representative.
- Each node has a pointer to the representative.
- Find-Set is $O(1)$
- Union is $O(n)$
- A sequence of $m$ operations is $O(m^2)$
- It would be a little better if we added shorter lists to the front of longer lists.
Outline

Motivation

Linked List Representation

Disjoint Set Forests

Heuristics

C Code using Arrays

Example
Rather than have each node keep a pointer to its representative, we keep a pointer to its parent. We can follow parent pointers up a tree to reach the representative at the root. We have no way to update each affected parent pointer when a union is done. We can simply attach the root of one tree below the root of the other.
Outline

Motivation

Linked List Representation

Disjoint Set Forests

Heuristics

C Code using Arrays

Example
Heuristics

- **Union by rank**
  - Keep the “rank” of each tree
  - Rank starts at 0
  - Increment the rank of a root when joining 2 nodes of equal rank
  - It is also easy to keep the size of each tree.

- **Path compression:**
  - During a find go through the path to the root twice.
  - On the second trip update all parent pointers to point directly to the root.

The effect of applying both heuristics is make the algorithms essentially linear time for a sequence of \( m \) union-find operations.
Outline

Motivation

Linked List Representation

Disjoint Set Forests

Heuristics

C Code using Arrays

Example
Use an array of ints $p$ to represent the forest.

Each element of the universe is an integer from 0 to $n - 1$.

$p[i]$ is the parent of $i$ if $p[i] \geq 0$.

$p[i]$ is negative if $i$ is a root node.

For a root node $p[i]$ is the size of the tree.
int find ( int p[], int x )
{
    int r, i;

    r = x;
    while ( p[r] >= 0 ) r = p[r];
    i = x;
    while ( p[i] >= 0 ) {
        x = p[i];
        p[i] = r;
        i = x;
    }

    return r;
}
void union ( int p[], int a, int b )
{
    int size;
    a = find ( p, a );
    b = find ( p, b );
    if ( a == b ) return;
    size = p[a] + p[b];
    if ( p[a] < p[b] ) {
        p[b] = a;
        p[a] = size;
    } else {
        p[a] = b;
        p[b] = size;
    }
}
Outline

Motivation

Linked List Representation

Disjoint Set Forests

Heuristics

C Code using Arrays

Example
Example

- Union(0,1), Union(2,3), ...
Example continued

Union(0,8), Union(1,11), Union(4,19)
Example continued

▶ Union(9,19)
Example continued

▶ Union(4,12)