p5q 1
MR

Data structures and algorithms 2003

(1) A million unit sets each containing distinct integers is to be successively combined by calls of `union(`S_1`, `S_2`)`. The result represents the union of the two disjoint sets represented by S_1 and S_2. Interspersed among these calls are several calls of `inSameSet` where `inSameSet(`m`, `n`)` yields `true` if and only if m and n are integers now in the same set. Describe in detail how you would implement `union` and `inSameSet` assuming they will be call about one and five million times, respectively. Explain by you solution is efficient. [10 marks]

(2) Describe in detail an implementation of Kruskal's algorithm for finding a minimum cost spanning tree of an undirected graph with positive integer costs on the edges that uses your version of `union` and `inSameSet`. [6 marks]

(3) Carefully prove that the spanning tree is unique if all the edge costs are distinct. [4 marks]

```
ANSWER NOTES:

Union-find material was added to the course this year.

(1) Bookwork.
Each set is represented by a tree of node.
Each node contains an integer in the set.
Each node is either the root in which case the integer is the representative
element for the set.
or it has a non null parent to another node.

UNION(s1, s2)  set the parent link of the representative node of one
to point to the representative node of the other. The resulting root node
should come from the larger set. Representative nodes should contain
the size of its set. Can use the parent link field plus one flag bit.

inSameSet(x, y) = TRUE if x and y have the same representative element.

Optimisation: always replace the parent link of any node visited
by a link to its representative node. This is called compression.
Later tests will be more efficient. If there are lots of calls of
inSameSet between calls of union, most nodes will be one edge away from
its root. So the cost of inSameSet will be almost O(1).
Union is O(1), increase the cost of inSameSet for the smaller set by
one for a while until compressions takes effect.

(2) Bookwork
Form singleton sets.
Repeated take the lowest cost edge connecting two previously disjoint
sets and unite the two sets. Continue until only one set left.

(3) Nearly bookwork.  First prove the algorithm finds a minimum cost
spanning tree (bookword) then point out that when all edges have
distinct cost the algorithm is deterministic since at the time i->j
is added, selecting any other edge would produce a tree that
has a strictly greater cost.

Suppose T is a minimum cost spanning tree with lower cost than S, the
```

tree found by Kruskal's algorithm. Sort the edges in T and S by
increasing cost. Let i->j be the first edge in S that is not in T.
The must be a path from i to j in T. Find the minimum cost edge p->q
on this path that is not so far not in S. Its cost will be strictly greater
that i->j. Delete p->q from T and add i->j. This will give a spanning
tree with lower cost which violates the initial assumption.