Analysis of Algorithms

Disjoint Sets

If more than one question appears correct, choose the more specific answer, unless otherwise instructed.

Concept: *disjoint sets as linked lists*

The following questions assume a linked-list implementation of a disjoint set. Assume each value has a pointer to its representative. Assume worst case behavior.

1. The *make-set* operation takes time:

   (A) constant
   (B) logarithmic
   (C) log linear
   (D) linear

2. The *find-set* operation takes time:

   (A) logarithmic
   (B) linear
   (C) quadratic
   (D) none of the other answers are correct
   (E) constant
   (F) log linear

3. Assuming no preference on which representative becomes the representative of the resulting set, the *union* operation takes time:

   (A) quadratic
   (B) linear
   (C) none of the other answers are correct
   (D) log linear
   (E) constant
   (F) logarithmic

4. Assuming the representative of the smaller set becomes the representative of the resulting set, the *union* operation takes time:

   (A) logarithmic
   (B) log linear
   (C) quadratic
   (D) linear
   (E) none of the other answers are correct
   (F) constant

5. Assuming the representative of the larger set becomes the representative of the resulting set, the *union* operation takes time:

   (A) linear
   (B) quadratic
   (C) constant
   (D) log linear
   (E) none of the other answers are correct
   (F) logarithmic

6. Assuming no preference on which representative becomes the representative of the resulting set, the total work of a series of m disjoint set operations is:

   (A) logarithmic
   (B) constant
   (C) linear
   (D) quadratic
   (E) none of the other answers are correct
   (F) log linear

7. Assuming the representative of the smaller set becomes the representative of the resulting set, the total work of a series of m disjoint set operations is:

   (A) quadratic
   (B) logarithmic
   (C) none of the other answers are correct
   (D) constant
   (E) log linear
   (F) linear

8. Assuming the representative of the larger set becomes the representative of the resulting set, the total work of a series of m disjoint set operations is:

(A) none of the other answers are correct
(B) logarithmic
(C) quadratic
(D) linear
(E) log linear
(F) constant

9. Assuming the representative of the larger set becomes the representative of the resulting set, how many times can a value's representative be updated in a series of union operations?

(A) quadratic
(B) logarithmic
(C) log linear
(D) constant
(E) none of the other answers are correct
(F) linear

10. Assuming the representative of the smaller set becomes the representative of the resulting set, how many times can a value's representative be updated in a series of union operations?

(A) constant
(B) logarithmic
(C) log linear
(D) quadratic
(E) linear
(F) none of the other answers are correct

11. Assuming no preference on which representative becomes the representative of the resulting set, how many times can a value's representative be updated in a series of union operations?

(A) log linear
(B) quadratic
(C) linear
(D) logarithmic
(E) none of the other answers are correct
(F) constant

Concept: *disjoint sets as trees*

The following questions assume a tree implementation of a disjoint set. Assume each value has a pointer to its parent with the root of the tree serving as the representative of the set. Assume worst case behavior.

12. **T** or **F**: A *union* operation runs in constant time if representatives are passed as arguments.

13. **T** or **F**: When the *union* operation is called, it always makes one of the arguments a child of the other.

14. **T** or **F**: When the *find-set* operation is called with path compression, the parent pointers of all nodes in the tree are set to the root.

15. **T** or **F**: When the *find-set* operation is called with path compression, the parent pointers of all nodes in the path from the node to the root are set to the root.

16. **T** or **F**: When the *find-set* operation is called with path compression, at most one parent pointer is set to the root.

17. How long does it take to perform m *make-set* operations?

(A) $theta(m \log m)$
(B) $theta(m^2)$
(C) $theta(1)$
(D) $theta(m)$

18. How long does it take to perform m *union* operations, assuming that representatives are passed as arguments?

(A) $theta(m)$
(B) $theta(m \log m)$
(C) $theta(1)$
(D) $theta(m^2)$

19. How long does it take to perform m *union* operations? Assume up-trees and union-by-rank.

(A) $theta(1)$
(B) $theta(m \log m)$
(C) $theta(m^2)$
(D) $theta(m)$

20. How many trees will there be after m *make-set* operations, starting with an empty disjoint set?

(A) m^2
(B) m
(C) $2m$
(D) 2^m

21. Suppose there are n trees of height zero in a disjoint set, what is the worst case height of the set if they are all unioned together? Do not assume union by rank.

(A) n
(B) $\log n$
(C) 1
(D) $\frac{n}{2}$

22. Suppose there are n trees of height zero in a disjoint set, what is the worst case height of the set if they are all unioned together? Assume union by rank.

(A) n
(B) $\log n$
(C) $\frac{n}{2}$
(D) 1

23. The *make-set* operation takes time:

(A) linear
(B) constant
(C) log linear
(D) logarithmic

24. The *find-set* operation (no path compression and no union by rank) takes time:

(A) log linear
(B) linear
(C) quadratic
(D) logarithmic
(E) none of the other answers are correct
(F) constant

25. The *find-set* operation (with path compression but no union by rank) takes time:

(A) logarithmic
(B) constant
(C) none of the other answers are correct
(D) quadratic
(E) linear
(F) log linear

26. The *find-set* operation (with no path compression but with union by rank) takes time:

(A) none of the other answers are correct
(B) quadratic
(C) logarithmic
(D) constant
(E) linear
(F) log linear

27. Assuming no preference on which representative becomes the representative of the resulting set, the *union* operation takes time:

(A) logarithmic
(B) quadratic
(C) constant
(D) log linear
(E) linear
(F) none of the other answers are correct

28. Assuming the representative whose tree has the smaller rank becomes the representative of the resulting set, the *union* operation takes time:

(A) linear
(B) logarithmic
(C) log linear
(D) quadratic
(E) constant
(F) none of the other answers are correct

29. Assuming the representative whose tree has the larger rank becomes the representative of the resulting set, the *union* operation takes time:

(A) logarithmic
(B) log linear
(C) linear
(D) none of the other answers are correct
(E) constant
(F) quadratic

30. Assuming no path compression and no union by rank, the total work of a series of m disjoint set operations is:

(A) linear
(B) log linear
(C) none of the other answers are correct
(D) logarithmic
(E) quadratic
(F) constant

31. Assuming path compression but no union by rank, the total work of a series of m disjoint set operations is:

(A) linear
(B) log linear
(C) quadratic
(D) none of the other answers are correct
(E) logarithmic
(F) constant

32. Assuming no path compression but union by rank, the total work of a series of m disjoint set operations is:

(A) linear
(B) constant
(C) quadratic
(D) logarithmic
(E) none of the other answers are correct
(F) log linear

33. Assuming path compression and union by rank, the total work of a series of m disjoint set operations is:

(A) logarithmic
(B) constant
(C) linear
(D) none of the other answers are correct
(E) log linear
(F) quadratic

34. Path compression is used to speed up the average cost of which operation(s)?

(A) union and make-set
(B) find-set
(C) find-set and make-set
(D) union
(E) make-set, find-set, and union
(F) none of the other answers are correct
(G) union and find-set
(H) make-set

35. Union by rank is used to speed up the average cost of which operation(s)?

(A) make-set, find-set, and union
(B) union
(C) make-set
(D) union and find-set
(E) union and make-set
(F) find-set
(G) find-set and make-set
(H) none of the other answers are correct

36. **T** or **F**: A single find-set operation with path compression takes asymptotically longer than a single find-set operation without path compression, in the worst case.

37. Suppose there are initially n disjoint sets. If m union operations are performed, what is the fewest number of disjoint sets that remain? Assume n is a power of two and $n > m$.

(A) $n - 2m$
(B) $n - 2^m$
(C) $n - m$
(D) m

38. Path compression occurs in which operation? Assume an implementation of *union* that is always passed representatives, as in `union(find-set(a),find-set(b))`.

(A) *union*
(B) *make-set*
(C) none of the other answers are correct
(D) *find-set*

39. Union-by-rank occurs in which operation?

(A) *find-set*
(B) none of the other answers are correct
(C) *union*
(D) *make-set*

For the following set of questions, consider the following set of operations:

```
for each i in 0..6 do make-set(i)
union(1,2)
union(3,4)
union(5,6)
union(6,2)
union(3,2)
find-set(3)
```

assuming union-by-rank and *NO* path compression. When unioning two sets having the same rank, assume the root with the larger value becomes the root of the resulting set.

40. After the `union(6,2)` operation how many children does 6's representative have?

(A) 4
(B) 1
(C) 2
(D) 3

41. After the `union(3,2)` operation how many children does 3's representative have?

(A) 4
(B) 3
(C) 1
(D) 2

42. After the `find-set(3)` operation how many children does 3's representative have?

(A) 2
(B) 4
(C) 3
(D) 6

Considering the previously provided list of operations on disjoint sets, assume there is union by rank AND path compression. When unioning two sets having the same rank, assume the root with the larger value becomes the root of the resulting set.

43. After the `union(6,2)` operation how many children does 6's representative have?

(A) 3
(B) 5
(C) 4
(D) 2

44. After the `union(3,2)` operation how many children does 3's representative have?

(A) 4
(B) 2
(C) 3
(D) 5

45. After the `find-set(3)` operation how many children does 3's representative have?

(A) 5
(B) 4
(C) 2
(D) 3

For the following set of questions, consider the following set of operations:

```
for each i in 0..9 do make-set(i)
union(1,2);
union(3,4);
union(5,6);
union(7,8);
union(9,0);
union(4,1);
```

```
union(4,6);
union(7,9);
find-set(3);
find-set(1);
```

assuming union by rank and path compression. When unioning two sets having the same rank, assume the root with the larger value becomes the root of the resulting set.

46. How many disjoint sets remain?

(A) none of the other answers are correct
(B) 5
(C) 4
(D) 3
(E) 2
(F) 1

47. How many values have a root as parent?

(A) 3
(B) 4
(C) 2
(D) none of the other answers are correct
(E) 5
(F) 1