Disjoint sets, B-trees, hash functions

1. (CLRS 21.1-1) Let $G(V, E)$ be a graph with vertices $V = \{a, b, c, d, e, f, g, h, i, j, k\}$. Use the disjoint-set datastructure to decide, how many connected components $G$ has, if the edges are examined in the order

$$(d, i), (f, k), (g, i), (b, g), (a, h), (i, j), (d, k), (b, j), (d, f), (g, j), (a, e).$$

Do the same exercise, but apply path compression and the union by rank principles. How many steps did your algorithm take?

2. In a far-away country there are $n$ cities but no roads. How many roads does the government need to build to make sure that all citizens can travel between any two cities?

3. Insert the keys $1, 3, 6, 4, 9, 8, 5, 11, 10, 12$ in the hash-table $T$ of length $5$, resolve collisions by chaining. Use the $\text{mod } 5$ operation as hash-function, thus $h(k) = k \mod 5$. How can you decide if number $13$ is a key in this table? What is the worst-case running time for finding a key in $T$? For additional practice do exercise CLRS 11.2-2.

4. (CLRS 11.2-1) Suppose we use hash function $h$ to hash $n$ distinct keys into an array $T$ of length $m$. Assuming simple uniform hashing, what is the expected number of collisions?

5. (CLRS 18.1-3) Show all legal B-trees of minimum degree $2$ that represent $\{1, 2, 3, 4, 5\}$.

6. Insert keys $10, 20, 30, 40, 50, 5, 15, 25, 1, 2$ into a B-tree $B$ of order $2$. Delete keys $10, 20$ and $50$. Show the changes in the tree.