Disjoint Sets I – Idiomatic

January 11, 2019

Maintain a family of disjoint sets under repeated union operations, using data types native to your programming language.

This exercise has the following secondary aims: (i) To ensure that you can handle basic input and output, including basic input parsing. (ii) To ensure that you can communicate with the Code Judge infrastructure we use for checking programming assignments, and handing in via the LearnIt teaching platform. (This is not a mandatory assignment, so you exercise will never actually be graded.) (iii) In conjunction with the following exercise *Disjoint Sets II*, to make clear that there is a difference between "How do I do X in idiomatic Python/Java?" and "How do I do X efficiently?" Both questions are valid, and have good answers, but they are not the same question. This exercise is about the first question. The rest of the course is about the second question.

Description

Maintain a family of disjoint sets, initially the singletons

$$\{0\}, \{1\}, \ldots, \{n-1\}$$

under the following operations:

union The *union* operation takes two arguments s and t and creates the union of the two sets containing s and t. More precisely, if $s \in S$ and $t \in T$ with $S \neq T$ then S and T are removed from the family and replaced by the set $S \cup T$. (If $S = T$ then nothing happens.)

query The *query* operation takes two arguments s and t and determines if s and t belong to the same set. More precisely, if $s \in S$ and $t \in T$ then print `yes` if $S = T$ and print `no` if $S \neq T$.

Note that these operations maintain invariantly that the sets in the family are disjoint.

Input

The input starts with two nonnegative integers n, m, on the first line. We have $0 \leq n \leq 10^5$ and $0 \leq m \leq 10^5$. The integer n is the initial number of singletons. The following m lines are of the form "`u` s t" or "`q` s t"; you can assume $0 \leq s < n$ and $0 \leq t < n$.

Output

For each query, a single line with `yes` or `no` as described above.

```
4 9
q 0 1
u 0 1
q 0 1
u 1 2
q 1 2
q 0 3
u 2 3
q 2 3
q 0 3
```

Figure 1: Sample input

```
no
yes
yes
no
yes
yes
```

Figure 2: Sample output

Deliverable

DS1. Solve the problem in standard Python or Java. Both languages support a datatype for sets: Python has a builtin class called `set()` whose objects support `union()` and `in`. Java has the `Set` interface in the `java.util` library, which is implemented in a number of ways, such as `HashSet` or `TreeSet`. Using these classes, write a readable, short, beautiful, idiomatic solution without regard for efficiency. Hand it in as `Ds1.java` or `ds1.py`. For this deliverable—and only this!—you can use all the native language features you want, including standard library methods that you don't understand, streams, generators, list comprehension, Python dicts and list methods, etc., even though I don't see why you'd want to do this. This is the last time in this course that you can need the full language. Relish it.

There is a set of instances on CodeJudge for you to test.

Report. Hand in a very brief report as a PDF, extrapolating the running time of your program to instances larger than 10000, preferably by just finishing the skeleton in the appendix.

Your program needs to be highly polished—remove unnecessary code, remove commented-out code, make sure your variable names are short and sweet, like those the book. (For instance, *do* use one-letter variable names like `S` and `T` for sets, not `theset` and `theotherset`, nor `variable` or `whatever`.) On the other hand, don't spend time on writing long comments or docstrings. Your program should be not much longer than 60 lines of Java or 30 lines of Python.

You can write the report by hand and scan it, or grab the LaTeX skeleton for the report from the course repo, or anything in between. It should be very short but highly polished—spelling mistakes or grammatical errors are completely unacceptable in an academic context. You are welcome to hand in the report in Danish.

Disjoint Sets I, Report

by Alice Cooper[1]

Results

I solved this exercise in Python 3 / Java.

Program `ds1` uses Java's `HashSet` and fails on tests [· · ·] with error [· · ·] / passes all the instances on CodeJudge.[2]

The following table gives a rough estimate on the running time for larger instances, based on experiments and extrapolating observed behaviour for smaller instances on my machine.[3]

Size	Time for `ds1`
$n, m = 10^5$	13.5 s
$n, m = 10^6$	3h 20m
$n, m = 10^7$	ca 1 week (extrapolated)
$n, m = 10^8$	ca 1 day (extrapolated)
$n, m = 10^9$	ca 1 day (extrapolated)

[1] Complete the report by filling in your real name, filling in the parts marked [. . .] and changing other parts wherever necessary. (For instance, the numbers is the tables are nonsense right now.) Remove the sidenotes in your final hand-in.

[2] Replace by whichever library class you actually use, and whichever errors you generated.

[3] The numbers in the example table are completely made up and have to be replaced. If you *want* to be more precise, you may. If you *want* to give averages and standard deviations for your experiments (because you've learned that) yo may. If you *want* to present the data graphically instead of a table, you may. If you *want* to present the data for different n and m, you may.