Advanced Data Structures (ADS-MIRI): 4-Empirical Study of Union-Find

The goal of this assignment is to implement several variants of Union-Find and conduct an experimental study of their (average) performance.

You will check the performance of the various possible combinations (12) when you choose a strategy for unions and a strategy for path compression. The choices for the unions are: (1) **QU**: unweighted quick-union, (2) **UW**: union-by-size (a.k.a. union-by-weight), (3) **UR**: union-by-rank. The choices for path compression are: (1) **NC**: no compression, (2) **FC**: full path compression, (3) **PS**: path splitting, (4) **PH**: path halving.

Once you fix some size n, you create an initial UnionFind data structure with n blocks, each different element on its own block. You shall them consider the $m = \binom{n}{2}$ distinct pairs of elements (i, j) one at a time, in random order (but never repeating!) and make a union for each such pair. The process should finish when there is only one single block in the data structure. This should happen more or less when you have processed around $\Theta(n \log n)$ of the pairs—as you process more and more pairs, many of them will involve two elements (i, j) already in the same block. Along the processing of the pairs, you have to measure (every $\Delta \geq 1$ pairs processed) the following parameters:

1. Total path length (**TPL**): for every element in the UF measure its distance to its representative (root of the tree) and sum up all the distances.

2. Total pointer updates (**TPU**): for every element in the UF measure the number of pointers which would be updated during a `Find` starting at that point, using the current path compression heuristic. For **NC** this value is 0. For the other strategies the computation of this quantity can be made more efficient if we know the number of children of each root; for instance, for **FC** we have

$$\mathbf{TPU} = \mathbf{TPL} - \sum_{r \text{ is a root}} \text{number of children of } r.$$

An appropriate linear combination of the two costs will give us a measure of the cost of each heuristic. The first measure gives us the cost of following pointers, whereas the second measures the effort made by path compression. Suppose that we take the cost of following one pointer to be a unit cost, and the cost of updating a pointer to be some $\epsilon \geq 1$. For example, the total cost of **FC** would be $2\mathbf{TPL} + \epsilon\mathbf{TPU}$.

Repeat the experiment making unions with random pairs several times, say $T = 20$ times, and measure **TPL** and **TPU** for each value $N \in \{0, \Delta, 2\Delta, 3\Delta, \ldots\}$ of pairs processed, then obtain average **TPL** and **TPU** as a function of N, by averaging the results of different executions, and as long as there are at least some minimum number, say 5, of executions which reached that particular number of pairs.

This has to be done for every combination of union strategy and path compression strategy. It is actually more efficient (and reduces the variance when you compare!) to supply each of the T sequences of random pairs to all twelve UFs that we can consider and gather the data for each one.

Once the full suite of experiments has been executed and data has been gathered, you have to prepare a report.

1. Describe briefly the program to execute the experiments. Give full listings of the code as an appendix of your report. Explain how you are computing the **TPL** and **TPU** at each configuration of the Union-Find.

2. Describe briefly the experimental setup.

3. Provide plots sumarizing the results of the experiments. It is important to consider some reasonably large values of n, say $n = 1000$, $n = 5000$ and $n = 10000$ and for each n plot the evolution of **TPL** and TPU as a function of N (you need not to measure **TPL** and TPU at every possible value of N, only every Δ pairs processed). Normalize by dividing **TPL** and **TPU** by n, giving the average distance to its root of a random node, and the average number of pointers updated if we make a find of a random node. It will be useful that the plots also show the average value at which the processing of sequences stops—as we mentioned in passing this will be $\Theta(n \log n)$.

   Another important kind of plots to include are those in which you compare different heuristics, for some fixed n. Instead of reporting **TPL** or **TPU** (actually their normalizations) it's more fair to fix some reasonable value of ϵ, sat $\epsilon = 2$, and report total costs or average costs (dividing by n).

An extra bonus (not mandatory) is to conduct a similar empirical study of the actual execution times following similar steps as described above. Do not forget to remove the code to count operations from the implementation, since it might introduce small disturbances in the measurements.

We encourage you to use LaTeX to prepare your report. For the plots you can use any of the multiple packages that LaTeX has (in particular, the bundle TikZ+PGF) or use independent software such as gnuplot and then include the images/PDF plots thus generated into your document.

Submit your work using the FIB-Racó. It must consist of a zip or tar file containing all your source code, auxiliary files and your report in PDF format. Include a README file that briefly describes the contents of the zip/-tar file and gives instructions on how to produce an executable program and reproduce the experiments. The PDF file with your report must be called

YourLastName_YourFirstName-4-union-find.pdf, and the zip/tar file must be called YourLastName_YourFirstName-4-union-find.zip (or .tar).