UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

Advanced Data Structures

Computer Science Department

Union Find Data Structure: An empirical analysis

Author:
Alex Herrero

Professor:
Conrado Marínez

February 17, 2025

Contents

List of Figures

List of Tables

1 Introduction

Given a binary relation R such that is:

- Reflexive: aRa
- Symmetric: $aRb \Rightarrow bRa$
- Transitive: $aRb \wedge bRc \Rightarrow aRc$

we say that R provides a partition Π of A into disjoint equivalence classes. That is, $\Pi = \{A_1, \ldots, A_k\}$ is defined as follows:

- $\forall A_i, A_j \in \Pi, A_i \cap A_j = \emptyset \iff A_i \neq A_j$
- $A = \bigcup_{i=1}^{k} A_i$
- $a \equiv b \iff a, b \in A_i$ for some $A_i \in \Pi$.

Such idea was used in Computer Science by Galler and Fisher[GF64] as the **union-find data structure** which is a data structure that stores partition of a set into disjoint sets. In particular **union find** consist of two main operation:

- Find Operation: Given two elements $a, b \in A$ determine if $\exists A_i \in \Pi$ s.t $a, b \in A_i$.
- Union Operation: Given two elements $a \in A_i$ and $b \in A_j$, *merge* A_i and A_j. That is, the result of this operation to the partition Π will be a new partition Π' such that $\Pi' = (\Pi \setminus \{A_i, A_j\}) \cup (A_i \cup A_j)$.

2 Implementation

From now on we are going to assume that our set A is defined as $\{0, \ldots, n-1\}$ (and if it is not the case we can use a dictionary to map elements from A to that range).

2.1 Representation of the partition

As every subset defines an equivalence class it is equivalent to represent every $A_i \subseteq A$ with a single element $a \in A_i$ which is called the *representative* of A_i (every other element of A_i will be related to a by the properties of the binary relation previously defined). Hence, in our union-find data structure every element will either the representative of a certain subset of A or it will point to someone who is in the same subset (later on different techniques will be discussed to see which is the best element "to point" and what that means).

A union-find data structure consists of an array $v[0 : n-1]$ and every element $i \in A$ will either point to another element $j \in A$ such that they both belong to the same class (in that case $v[i] = j$) or i will be the representative of a certain class and it will be marked *specially*.

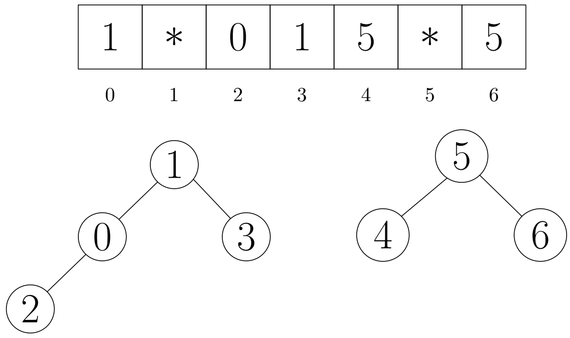

Figure 1: An example of a representation of a Union-Find

References

[GF64] Bernard A Galler and Michael J Fisher. An improved equivalence algorithm. *Communications of the ACM*, 7(5):301–303, 1964.