



Doctoral Thesis

Search processes and their average case analysis

Author(s):

Galli, Nicola

Publication Date:

1998

Permanent Link:

<https://doi.org/10.3929/ethz-a-001991004> →

Rights / License:

[In Copyright - Non-Commercial Use Permitted](#) →

This page was generated automatically upon download from the [ETH Zurich Research Collection](#). For more information please consult the [Terms of use](#).

Diss. ETH No. 12880

Search Processes and their Average Case Analysis

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZURICH

for the degree of
Dr. sc. techn.

presented by
NICOLA GALLI
Dipl. Informatik-Ing. ETH
born 28.5.1971
citizen of Besazio, Switzerland

accepted on the recommendation of
Prof. Dr. Klaus Simon, examiner
Prof. Dr. Emo Welzl, co-examiner

1998

Abstract

The last decade is characterized by a rapid growth of the amount of information stored worldwide. The role of search strategies is, therefore, a key aspect of the management of data. Many algorithms have been developed to obtain more and more efficient solutions where time and space complexity have been mostly considered to measure the efficiency.

The traditional type of complexity treatment is the so called *worst case analysis*. However, this approach is often rather pessimistic. The observed behavior is in many cases much better than the worst case. Therefore, it is meaningful to analyze the average performance. But this approach involves a new problem: A probabilistic model is needed. The major difficulty is to find a good compromise to get both a general and simple model.

In this thesis, we handle two basic problems from this point of view. First, we consider the search in graphs. Starting from an average case analysis of Breadth-First-Search, we propose a general model to investigate a vast class of graph algorithms. This model is also useful to analyze structural properties of random graphs. Then, we consider the search in ordered sets, namely the dictionary problem. We study two different solutions: Randomized search trees and Hashing. More precisely, we present a realization of randomized search trees which extends the traditional solution [6] in the weighted case and which points out the affinity with balanced search trees. Then, we propose a perfect hashing scheme based on the compression of sparse tables. The resulting perfect hash function is surprisingly simple and efficient (in terms of time and space complexity).

Riassunto

L'ultimo decennio è stato caratterizzato da una rapida crescita del volume di informazioni memorizzate in tutto il mondo. Il ruolo delle strategie per la ricerca dei dati è quindi divenuto un aspetto determinante per la loro gestione. Molti algoritmi sono stati sviluppati per ottenere soluzioni sempre più efficienti.

Tradizionalmente gli studi riguardanti l'efficienza di un algoritmo – misurata soprattutto in termini di tempo e di memoria impiegati – sono basati sull'analisi del caso peggiore (*worst case analysis*). Però i risultati si sono rilevati spesso pessimistici se confrontati con valori empirici. Di conseguenza si è pensato di analizzare il comportamento medio di un algoritmo (*average case analysis*). Purtroppo anche questo approccio non è privo di problemi: infatti richiede un modello probabilistico. La difficoltà maggiore consiste nel trovare un modello generale e semplice allo stesso tempo.

In questa tesi trattiamo due problemi in quest'ottica. Per prima cosa studiamo l'esplorazione di grafi. Partendo dall'analisi di uno degli algoritmi base, vale a dire *Breadth-First-Search*, proporremo un modello generale per analizzare una vasta classe di algoritmi su grafi. Questo modello permette inoltre uno studio della struttura di grafi casuali. In seguito considereremo la ricerca di dati in insiemi ordinati. Studieremo due soluzioni praticamente opposte: strutture ad albero randomizzate e hashing. Nel primo caso presenteremo una realizzazione che estende la soluzione tradizionale [6] e che mostra le affinità con alberi bilanciati (*balanced trees*). Infine proporremo un metodo di hashing basato sulla compressione di matrici sparse. Il risultato è sorprendentemente semplice ed efficiente (sia dal punto di vista del tempo che della memoria).