Fast Dynamic Arrays
We present a highly optimized implementation of tiered vectors, a data structure for maintaining a sequence of n elements supporting access in time $O(1)$ and insertion and deletion in time $O(n)$ for $> 0$ while using $o(n)$ extra space. We consider several different implementation optimizations in C++ and compare their performance to that of vector and set from the standard library on sequences with up to 108 elements. Our fastest implementation uses much less space than set while providing speedups of 40× for access operations compared to set and speedups of 10.000× compared to vector for insertion and deletion operations while being competitive with both data structures for all other operations.

General information
State: Published
Organisations: Department of Applied Mathematics and Computer Science, Algorithms and Logic
Authors: Bille, P. (Intern), Christiansen, A. R. (Intern), Ettienne, M. B. (Intern), Gørtz, I. L. (Intern)
Pages: 16:1--16:13
Publication date: 2017

Time-space trade-offs for lempel-ziv compressed indexing
Given a string S, the compressed indexing problem is to preprocess S into a compressed representation that supports fast substring queries. The goal is to use little space relative to the compressed size of S while supporting fast queries. We present a compressed index based on the Lempel-Ziv 1977 compression scheme. Let n, and z denote the size of the input string, and the compressed LZ77 string, respectively. We obtain the following time-space trade-offs. Given a pattern string P of length m, we can solve the problem in (i) $O(m + occ \lg lg n)$ time using $O(z \lg (n/z) \lg lg z)$ space, or (ii) $O((m + lg z/lg(n/z) + occ(lg n + lg z))$ time using $O(z \lg (n/z))$ space, for any $0 < \delta < 1$. In particular, (i) improves the leading term in the query time of the previous best solution from $O(m \lg m)$ to $O(m)$ at the cost of increasing the space by a factor $lg lg z$. Alternatively, (ii) matches the previous best space bound, but has a leading term in the query time of $O(m(1 + lg z/lg(n/z)))$.

General information
State: Published
Organisations: Department of Applied Mathematics and Computer Science, Algorithms and Logic, Department of Informatics and Mathematical Modeling
Reimplementing a Multi-Agent System in Python

We provide a brief description of our Python-DTU system, including the overall design, the tools and the algorithms that we used in the Multi-Agent Programming Contest 2012, where the scenario was called Agents on Mars like in 2011. Our solution is an improvement of our Python-DTU system from last year. Our team ended in second place after winning at least one match against every opponent and we only lost to the winner of the tournament. We briefly describe our experiments with the Moise organizational model. Finally we propose a few areas of improvement, both with regards to our system and to the contest.

General information
State: Published
Organisations: Department of Applied Mathematics and Computer Science, Algorithms and Logic, Technical University of Denmark
Authors: Villadsen, J. (Intern), Jensen, A. S. (Intern), Ettienne, M. B. (Intern), Vester, S. (Intern), Andersen, K. B. (Ekstern), Frøsig, A. (Ekstern)
Pages: 67-84
Publication date: 2013

Host publication information
Title of host publication: The Multi-Agent Programming Contest 2012 Edition Evaluation and Team Descriptions
Publisher: Technische Universität Clausthal
Editors: Köster, M., Schlesinger, F., Dix, J.
Number: Ifi-13-01
ISSN: 1860-8477
Main Research Area: Technical/natural sciences
Links:
http://www.in.tu-clausthal.de/fileadmin/homes/techreports/ifi1301koester.pdf
Publication: Research - peer-review › Report chapter – Annual report year: 2013

Reimplementing a Multi-Agent System in Python

We provide a brief description of our Python-DTU system, including the overall design, the tools and the algorithms that we used in the Multi-Agent Programming Contest 2012, where the scenario was called Agents on Mars like in 2011. Our solution is an improvement of our Python-DTU system from last year. Our team ended in second place after winning at least one match against every opponent and we only lost to the winner of the tournament. We briefly describe our experiments with the Moise organizational model. Finally we propose a few areas of improvement, both with regards to our system and to the contest.

General information
State: Published
Implementing a Multi-Agent System in Python

We describe the solution used by the Python-DTU team in the Multi-Agent Programming Contest 2011, where the scenario was called Agents on Mars. We present our auction-based agreement algorithm and discuss our chosen strategy and our choice of technology used for implementing the system. Finally, we present an analysis of the results of the competition as well as propose areas of improvement.

General information
State: Published
Organisations: Department of Applied Mathematics and Computer Science, Algorithms and Logic, Department of Informatics and Mathematical Modeling, Algorithms and Logic
Authors: Ettienne, M. B. (Intern), Vester, S. (Intern), Villadsen, J. (Intern)
Pages: 147-179
Publication date: 2012
Multi-Agent Programming Contest 2012 - The Python-DTU Team

We provide a brief description of the Python-DTU system, including the overall design, the tools and the algorithms that we plan to use in the agent contest.

General information
State: Published
Organisations: Department of Informatics and Mathematical Modeling, Algorithms and Logic, Department of Applied Mathematics and Computer Science, Algorithms and Logic, Technical University of Denmark
Authors: Villadsen, J. (Intern), Jensen, A. S. (Intern), Ettienne, M. B. (Intern), Vester, S. (Intern), Balsiger Andersen, K. (Ekstern), Frøsig, A. (Ekstern)
Number of pages: 4
Publication date: 2012

Publication information
Publisher: Technical University of Denmark (DTU)
Original language: English
Main Research Area: Technical/natural sciences
Electronic versions:
1210.0437v1.pdf
Links:
http://arxiv.org/pdf/1210.0437v1
http://multiagentcontest.org/
Source: dtu
Source-ID: u::6505
Publication: Research - peer-review › Report – Annual report year: 2012

Multi-Agent Programming Contest 2011 - The Python-DTU Team

We provide a brief description of the Python-DTU system, including the overall design, the tools and the algorithms that we plan to use in the agent contest.

General information
State: Published
Organisations: Algorithms and Logic, Department of Informatics and Mathematical Modeling, Department of Applied Mathematics and Computer Science, Algorithms and Logic
Authors: Villadsen, J. (Intern), Ettienne, M. B. (Intern), Vester, S. (Intern)
Number of pages: 4
Publication date: 2011

Publication information
Publisher: Technical University of Denmark (DTU)
Original language: English
Main Research Area: Technical/natural sciences
Electronic versions:
Projects:

**Compressed Computation on Structured Data**
Department of Applied Mathematics and Computer Science  
Period: 01/05/2015 → 13/08/2018  
Number of participants: 3  
Phd Student:  
Ettienne, Mikko Berggren (Intern)  
Supervisor:  
Gørtz, Inge Li (Intern)  
Main Supervisor:  
Bille, Philip (Intern)

**Financing sources**  
Source: Internal funding (public)  
Name of research programme: Samfinansieret - Andet  
Project: PhD