ACM ICPC CONTENTS

	ICPC Training Material: Data Structures,			2.2.3 Z Algorithm	4
	Algorithms and Theorems		2.2	2.2.4 Aho Corasick Algorithm	4
			2.3	Graph Algorithms	4
Contents				2.3.2 Depth First Search	4
1	Data Structures	2		2.3.3 Djikstra's Algorithm	4
	1.1 Elementary Data Structures	2		2.3.5 Prim's Algorithm	4
	1.1.1 Array	2		2.3.6 Kruskal's Algorithm	4
	1.1.2 Linked List	2		2.3.7 Topological Sort	7
	1.1.3 Stack	2		2.3.8 Johnson's algorithm	7
	1.1.4 Queue	3	2.4		7
	1.1.5 Heap	3	2.4	2.4.1 Ford Fulkerson Algorithm	
	1.1.6 Hash	3		2.4.2 Dinic's Algorithm	2
	1.2 Advanced Data Structures	3		2.4.3 Hopcroft Karf Algorithm	2
	1.2.1 Priority queues	3		2.4.4 Gomory-Hu Algorithm	4
	1.2.2 Fenwick Tree	3		2.4.5 Stoer-Wagner Algorithm	4
	1.2.3 K-D Tree	3	2.5	Geometrical Algorithms	4
	1.2.4 Interval Tree	3		2.5.1 Convex hull Algorithm	4
	A1 - 11			2.5.2 Graham scan Algorithm	4
2	Algorithms 2.1 Sorting and Searching	3		2.5.3 Bentley-Ottmann Algorithm	4
	2.1 Sorting and Searching	3		2.5.4 Rotating calipers	4
	2.1.2 Quick Sort	3			
	2.1.3 Merge Sort	³ 3	Ma	thematics	4
	2.1.4 Heap Sort	4	3.1	Number Theory	4
	2.1.5 Intro Sort	4		3.1.1 Lucas Theorem	4
	2.2 String manipulation	4		3.1.2 Chinese remainder Theorem	4
	2.2.1 KMP Algorithm	4		3.1.3 Primality test	4
	2.2.2 Rabin Karp	4		3.1.4 Sieve of Eratosthenes	4

ACM ICPC 1 DATA STRUCTURES

1 Data Structures

1.1 Elementary Data Structures

In Computer Science, in order to treat and store data, it first needs to be structured. Hence, multiple data structures were created: Array, Hash, Queue, Tree and multiple others.

1.1.1 Array

The array is the most used data structure. It consists on a collection of values, such as each value is identified by at least one index.

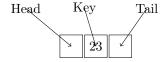


Arrays are useful because they exploit the addressing logic of computers. Generally, the memory is a one-dimensionnal array of words, whose indices are the addresses.

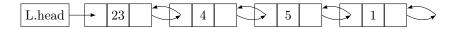
1.1.2 Linked List

The linked list is a linear data structure in which the objects are ordered according to the value of their pointer.

Every object x of the linked list L has a key attribute x.key and a successor x.succ attribute.



In the case of double linked lists, a x.prec attribute is added. An exemple is given below :



In the case of a circular list, the attribute x.prec at the queue of the list points at the head, and the attribute x.succ at the head of the list points at the tail. Hence, the list can be seen as a **ring** of elements.

Linked List provide the advantage of having a dynamic size, as in an rray memory is allocated during compilation time while it is allocated at runtime for linked lists.

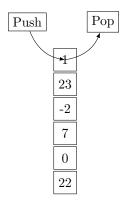
1.1.3 Stack

The stack consists on a collection of data that is added and retrieved according to the FILO(First In Last Out) method.

Two operations, which can only be applied on the top element of the stack, are used to manipulate the data :

- Push: Used to add a block on the top of the stack. A stack is overflowed if the number of blocks exceeds the capacity.
- Pop: Used to retrieve the block at the top of the stack.

ACM ICPC 2 ALGORITHMS



- 1.1.4 Queue
- 1.1.5 Heap
- 1.1.6 Hash
- 1.1.7 Trees
- 1.2 Advanced Data Structures
- 1.2.1 Priority queues
- 1.2.2 Fenwick Tree
- 1.2.3 K-D Tree
- 1.2.4 Interval Tree

2 Algorithms

- 2.1 Sorting and Searching
- 2.1.1 Binary Search
- 2.1.2 Quick Sort

The main concept behind this algorithm is the **divide and reign** principle. At each iteration, the initial array is progressively divided into a bigger number of sub-arrays. Each sub-array is sorted by choosing a **pivot** and placing the elements smaller than him at his left.¹

Here is a pseudo-code scheme is attributed to $\it Nico\ Lomuto$:

- 1: function QUICKSORT(A, lo, hi)
- 2: $p \leftarrow \text{partition}(A, \text{lo,hi})$

also used for memory management, as well as in expression evaluation. Stacks are implemented using arrays or linked lists, the two previous data structures.

An intuitive application would be inversing strings or numbers. It is

¹Hence, the bigger elements are placed on his right.

ACM ICPC 3 MATHEMATICS

```
QUICKSORT(A, lo, p - 1)
                                                                       2.1.3
                                                                              Merge Sort
 3:
      QUICKSORT(A, p + 1, hi)
 4:
                                                                              Heap Sort
                                                                       2.1.4
 5: end function
 6:
                                                                       2.1.5
                                                                              Intro Sort
 7: function Partition(A, lo, hi)
                                                                       2.2
                                                                             String manipulation
 8:
      i \leftarrow lo
      for j \leftarrow lo \text{ to } hi - 1 \text{ do}
 9:
                                                                              KMP Algorithm
                                                                       2.2.1
         if A[j] \leq A[hi] then
10:
                                                                       2.2.2
                                                                              Rabin Karp
             Switch A[i] with A[j]
11:
12:
             i \leftarrow i + 1
                                                                              Z Algorithm
                                                                       2.2.3
          end if
13:
                                                                       2.2.4
                                                                              Aho Corasick Algorithm
          j \leftarrow j + 1
14:
      end for
15:
                                                                       2.3
                                                                             Graph Algorithms
      Switch A[i] with A[hi] return i
16:
17: end function
                                                                       2.3.1
                                                                              Breadth First Search
                                                                       2.3.2
                                                                              Depth First Search
                                                                              Djikstra's Algorithm
                                                                       2.3.3
                                                                       2.3.4
                                                                              Floyd Warshall's Algorithm
                                                                       2.3.5
                                                                              Prim's Algorithm
                                                                       2.3.6
                                                                              Kruskal's Algorithm
                                                                       2.3.7
                                                                              Topological Sort
                                                                              Johnson's algorithm
                                                                       2.3.8
                                                                             Network Flow Algorithms
  Quicksort can be easily parallelized, and the performance on parallel
quicksort is better than mergesort and heapsort. However, it has the
                                                                       2.4.1
                                                                              Ford Fulkerson Algorithm
possibility to degenerate to O(n^2), which can be devastating if used in
                                                                              Dinic's Algorithm
                                                                       2.4.2
large data sets.
                                                                              Hopcroft Karf Algorithm
                                                                       2.4.3
                                                                     ^{4} 2.4.4
                                                                              Gomory-Hu Algorithm
                                                                              Stoer-Wagner Algorithm
                                                                       2.4.5
                                                                       2.5
                                                                             Geometrical Algorithms
```

Convex hull Algorithm
Graham scan Algorithm