1 Runtime

$$T^{\text{worst}}(n) = \max_{\forall I} T(I)$$

$$T^{\text{avg}}(n) = \frac{1}{n!} \sum_{\forall \pi} T(\pi)$$

$$T^{\text{exp}}(I) = \sum_{R} T(I,R)P(R)$$

$$T^{\text{worst exp}}(n) = \max_{\forall I} T^{\text{exp}}(I)$$

2 Useful formulas

$$\sum_{i=0}^{n-1}(a+di) \in \Theta(n^2), d \neq 0$$

$$\sum_{i=0}^{n-1} ar^i \in \begin{cases} \Theta(r^{n-1}) & r > 1 \\ \Theta(n) & r = 1 \\ \Theta(1) & r < 1 \end{cases}$$

$$\sum_{i=1}^{n} \frac{1}{i} \in \Theta(\log n)$$

$$\sum_{i=1}^{\infty} \frac{1}{i^2} \in \Theta(1)$$

$$\sum_{i=1}^{\infty} \frac{i}{2^i} \in \Theta(1)$$

$$\sum_{i=1}^{n} i^k \in \Theta(n^{k+1}), k \geq 0$$

3 Order

O	$\exists c \exists k f(n) \leq cg(n)$
Ω	$\exists c \exists k f(n) \geq cg(n)$
Θ	$\exists c_1, c_2 \exists k c_1 g(n) \leq f(n) \leq c_2 g(n)$
o	$\forall c \exists k f(n) \leq cg(n)$
ω	$\forall c \exists k f(n) \geq cg(n)$

3.1 Transitivity

O, O	O
Ω, Ω	Ω
O, o	o

3.2 Limit theorem

$$L = \lim_{n \to \infty} \frac{f(n)}{g(n)}$$

$L = 0$	o
$L \in (0, \infty)$	Θ
$L = \infty$	ω

3.3 Recurrence

$T(n/2) + O(1)$	$O(\log n)$
$2T(n/2) + O(n)$	$O(n \log n)$
$2T(n/2) + O(\log n)$	$O(n)$
$cT(n-1) + O(1), c < 1$	$O(1)$
$2T(n/4) + O(1)$	$O(\sqrt{n})$
$T(\sqrt{n}) + O(\sqrt{n})$	$O(\sqrt{n})$
$T(\sqrt{n}) + O(1)$	$O(\log \log n)$

4 Heap

Heapify	$O(n)$
Insert	$O(\log n)$
DeleteMax	$O(\log n)$
FixUp/Down	$O(\log n)$
HeapSort	$O(n \log n)$

Structural property: all levels are full except possibly last level, which is left-justified

Heap-order property: parent key not less than child key (max heap)

Height: $h \in O(\log n)$

Parent: $\lfloor (i-1)/2 \rfloor$

Left child: $2i + 1$

Right child: $2i + 2$

5 Selection and sort

HoarePartition	$O(n)$
QuickSelect	$O(n)/O(n^2)$
QuickSort	$O(n \log n)/O(n^2)$

6 Decision tree

Comparison-based sorting algorithm: $\Omega(n \log n)$

Comparison-based search: $\Omega(\log n)$

7 Non-comparison based sort

BucketSort	$O(n + R)$
MSDRadixSort	$O(mnR)$
LSDRadixSort	$O(m(n + R))$

8 AVL tree

zxy: sequence of tallest children from z

R	$R(z)$
L	$L(z)$
LR/DR	$L(y), R(z)$
RL/DL	$R(y), L(z)$

9 Skip list

$$P(X_k \geq i) = \left(\frac{1}{2}\right)^i$$

$$P(X_k = i) = \left(\frac{1}{2}\right)^{i+1}$$

10 Ordering

MTF: move to front (2-competitive)

Transpose: swap with next

Freq-count: sort based on access counter

11 Interpolation search

$$m = l + \left\lfloor \frac{k - A[l]}{A[r] - A[l]}(r - l) \right\rfloor$$

12 Trie

$O(|w|)$ for all operations

Compress trie has up to $2n - 1$ nodes.

12.1 Hashing

Direct addressing: no hash

Chaining: linked list for collision

Linear: $h(k,i) = h_0(k) + i$

Double: $h(k,i) = h_0(k) + ih_1(k)$

Cuckoo: two tables, evict to other

13 Range search

$$\rho(S) = \frac{L}{d_{\min}}$$

Quadtree: $h \in \Omega(\log \rho(S))$, operations $\Theta(h)$

kd-tree: $O(n \log n)$ build, $O(s + n^{1-\frac{1}{d}})$ query, $O(n)$ space

Range tree: $O(n(\log n)^d)$ build, $O(s + (\log n)^d)$ query, $O(n \log n^{d-1})$ space

14 String matching

Brute force: $O(mn)$ search

KR: $O(m)$ pre, $O(n+m)$ search, $O(mn)$ worst, $O(1)$ space

KMP: $O(m)$ pre, $O(n)$ search, $O(m)$ space

BM: $O(m + |\Sigma|)$ pre, $O(n + |\Sigma|)$ search, $O(m + |\Sigma|)$ space

Suffix tree: $O(|\Sigma|\, n)$ pre, $O(m\,|\Sigma|)$ search, $O(n)$ space

Suffix array: $O(n \log n)$ pre, $O(m \log n)$ search, $O(n)$ spac

15 Compression

15.1 Huffman

Start with a trie for every character and its frequency. Merge tries with smallest frequencies together.

15.2 RLE

Encode lengths of runs of same character. Use prefix-free encoding of length.

15.3 LZW

Start with initial dictionary D of all characters.

15.4 BWT

16 Distributions

Dist.	$P(X = x)$	$E(X)$
Binomial	$\binom{n}{x} p^x q^{n-x}$	np
Geometric	pq^x	q/p