Sort

Insertion Sort	$O(n^2)$	$O(n)$	
Counting Sort	$O(k+n)$	$O(n)$	conoscendo il range di valori
Heap Sort	$O(n\log(n))$	——	
Quick Sort	$O(n\log(n))$	$O(n^2)$	nel caso di vettore giá ordinato

Grafo

BFS - DFS	$O(n+m)$	——	
Componenti Connesse	$O(n+m)$	——	
Strong Componenti Connesse	$O(n+m)$	——	
Ciclo	$O(n+m)$	——	

Cammini Minimi

Dijkstra (coda a prioritá)	$O(n^2)$	deleteMin() $\forall$ nodo $\Rightarrow O(n)n$
Johnson (coda a prioritá con heap binario)	$O(m\log(n))$	migliore per grafi sparsi, m = num archi
Belman-Ford-Moore (coda)	$O(n+m)$	il migliore perché lavora con archi negativi

Cammini Minimi fra tutte le Coppie di Nodi

Floyd-Warshall	$O(n^3)$	

Albero di Copertura Minima

Kruskal	$O(m\log(n))$	ordina archi in modo crescente di peso e utilizza MFSET
Prim	$O(m\log(n))$	aggiunge sempre l'arco con peso minore partendo da un nodo radice

Rete di Flusso

Ford-Fulkerson	$O(\lvert f^\star \rvert\,(m+n))$	cammino aumentante con DFS che csta $O(n+m)$
Edmond-Kapp	$O(nm^2)$	cammini aumentanti con BFS

Master Theorem

$$T(n) = \sum_{1 \leq i \leq h} a_i T(n-i) + cn^b$$

ponendo $\alpha = \sum_{1 \leq i \leq h} a_i$ si ottiene:

1. $T(n)$ é $O(n^{b+1})$ se $a = 1$
2. $T(n)$ é $O(a^n n^b)$ se $a \geq 2$

$$T(n) = aT(n/b) + cn^\beta$$

ponendo $\alpha = \frac{\log(a)}{\log(b)}$ si ottiene:

1. $T(n)$ é $O(n^\alpha)$ se $\alpha > \beta$
2. $T(n)$ é $O(n^\alpha \log(n))$ se $\alpha = \beta$
3. $T(n)$ é $O(n^\beta)$ se $\alpha < \beta$

Lista con Puntatori

costruzione	$O(1)$	
insert	$O(1)$	
remove	$O(1)$	
get	$O(n)$	
exists	$O(n)$	

Stack e Queue

costruzione	$O(1)$	
push, enqueue	$O(1)$	solo in testa
pop, top	$O(1)$	solo in testa
top, dequeue	$O(1)$	
isEmpty	$O(1)$	

Tabelle Hash - con ABR

costruzione	$O(1)$	
insert	$O(\log(n))$	
lookup	$O(\log(n))$	
remove	$O(\log(n))$	

Alberi Binari di ricerca

costruzione	$O(1)$	
insertNode	$O(\log(n))$	
lookupNode	$O(\log(n))$	
removeNode	$O(\log(n))$	

Grafo con Matrice di Adiacenza

costruzione	$O(1)$	
adj	$O(n^2)$	
V	$O(n)$	
spazio di memoria	$O(n^2)$	
visita	$O(n+m)$	

Grafo con Liste di Adiacenza

costruzione	$O(1)$	
adj	$O(n)$	
V	$O(n)$	
spazio di memoria	$O(n)$	
visita	$O(n+m)$	

Merge Find Set - basata su Foresta con compressione dei cammini ed Euristica sul Rango

costruzione	$O(n)$	
merge	$O(1)$	
find	$O(1)$	