COMP20007/Design of Algorithms S1 2013 Summary

(Note: these are the bounds/algorithms we have learnt in this course: there may be more efficient ones!)

Algorithm	Best Case	Worst Case	Reason Studied	Notes
Fibonacci–D&C	$O(2^n)$	$O(2^n)$	Simple Divide and Conquer	Size of $f(n)$ must fit in word
Fibonacci–DP	$O(n)$	$O(n)$	Simple DP conversion	
Mergesort	$O(n \log n)$	$O(n \log n)$	Conquer heavy Useful (parallel/clusters) Intro to recurrence equations	Induction Master Theorem
Quicksort	$O(n)$ if 3-way part	$O(n^2)$	Divide heavy Partitioning Bad worst case	Best = all equal Average case? $O(n \log n)$
Median Finding	$O(n)$	$O(n)$	Partitioning Analysis	Hoare - like quick-sort can be $O(n^2)$ Random pivot - on average need $O(n)$. $E = \frac{1}{2}+\frac{1}{2}(1+E) \to E = 2$ BFPRT - median-of-median pivot $O(n)$
Strassen's MM	$O(n^{2.81})$	$O(n^{2.81})$	Subtleties of D-and-C and analysis using Mast. Theorem	7 rather than 8 in $T(n) = 7T(n/2) + O(n^2)$
Radix Sort	$O(nk)$	$O(nk)$	LTS, useful Partitioning	Fast $k = \lceil \log_2(\max\{A\}) \rceil$
Counting Sort	$O(n+K)$	$O(n+K)$	LTS, useful	$K = \max\{A\}$

Algorithm	Best Case	Worst Case	Reason Studied	Notes
Graph DFS (AL)	$O(n+m)$	$O(n+m)$	Intro to graphs Connectedness problems	Pre and post numbers Edge classification Stack based
Graph DFS (AM)	$O(n^2)$	$O(n^2)$		Generally: $E = O(V^2)$
Top. Sort	$O(n+m)$	$O(n+m)$	Dags and dependencies Sources and Sinks	
Graph BFS (AL)	$O(n+m)$	$O(n+m)$	Useful for paths	Queue based
Graph BFS (AM)	$O(n^2)$	$O(n^2)$		
Connect. Comp.	$O(n+m)$	$O(n+m)$	Intro the reverse of G	Power of DAGs, Based on BFS
Dijkstra's SSSP (AL + Heap)	$O((n+m)\log n)$	$O((n+m)\log n)$	Data structures are important Useful	
Dijkstra's SSSP (AM + USA)	$O(nm)$	$O(nm)$		If heap, $O(nm \log n)$
Bellman Ford SSSP	$O(nm)$	$O(nm)$	Part of Dijkstra's repeated	
Kruskal (LL + head ptr)		$O(m \log n)$	Greedy	$E = O(V^2)$
Kruskal (Forest with PC)		$O(m \log^* n)$	How DS can help Amortised Analysis Meet $\log^* n$	Assumes counting sort on E

Algorithm	Best Case	Worst Case	Reason Studied	Notes
Prim-Jarnik		$O((m+n)\log n)$	Alternate greedy for MST No disjoint-set required	
Set Cover	$O(m+n)$	$O(nm)$	Greedy Approximation	NP-Complete
Unary code build		$O(1)$	Intro to codes	
Huffman		$O(n\log n)$	Optimal prefix code Greedy	Assume input unsorted.
Shannon-Fano		$O(n\log n)$	D&C approach to code generation	
BWT Construction		$O(n^2\log n)$	Suffix algorithm Combine with MTF Can search in it	
Pattern search (Brute force)		$O(nm)$	Intro to problem	`grep`
Knuth Morris Pratt		$O(n+m)$	Standard (eg Ctrl-F)	Not studied in detail
Rank (store all)		$O(1)$	Foundation of many succinct DS	
Rank (Wavelet Tree)		$O(\log\sigma)$	Intro to Wavelet Tree	
Pattern search BWT + space!	$O(m)$	$O(m)$		If use lots of space for `rank`
Pattern search BWT+Wavelet Tree	$O(m\log\sigma)$	$O(m\log\sigma)$		Amazing! Compressed space (Huffman shaped WT)
Edit Distance		$O(nm)$	Intro Dyn. Prog.	

Algorithm	Best Case	Worst Case	Reason Studied	Notes
Long. inc. subseq.		$O(n^2)$	DP eg on graphs	Can be done in $O(n \log n)$ but not studied.
Knapsack		$O(nW)$	DP and Greedy eg	NP-Complete
Pretty-print		$O(n^2)$	DP eg	
SAT		$O(2^n)$	Fundamental to comp. theory	NP-Complete
TSP		$O(n!)$		NP-Complete

Data Structure	Insert	Delete	Find with map	Find no map	Successor with map	Find Min.
Unsort. Arr.	$O(1)$	$O(n)$	$O(1)$	$O(n)$	$O(n)$	$O(n)$
Sorted Arr.	$O(n)$	$O(n)$	$O(1)$	$O(\log n)$	$O(1)$	$O(1)$
Uns. LL	$O(1)$	$O(n)$	$O(1)$	$O(n)$	$O(n)$	$O(n)$
S LL	$O(n)$	$O(n)$	$O(1)$	$O(n)$	$O(1)$	$O(1)$
Heap	$O(\log n)$	$O(\log n)$	$O(1)$	$O(n)$	$O(1)$	$O(1)$
BST	$O(n)$	$O(n)$	$O(1)$	$O(n)$	$O(\log n)$?	$O(n)$
AVL tree	$O(\log n)$	$O(\log n)$	$O(1)$	$O(\log n)$	$O(\log n)$	$O(\log n)$
Hash table (assuming good HF)	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(m)$	$O(m)$

Disjoint set	Make	Union	Find with map	Find without map
LL with head ptr	$O(1)$	$O(n)$. $O(n \log n)$ over sequence.	$O(1)$	$O(n)$
Forest no PComp	$O(1)$	$O(1)$	$O(\log n)$	Impossible
Forest with PComp	$O(1)$	$O(\log^* n)$	$O(\log^* n)$	Impossible