

Binary search:

Skiptir hlutum í tvö hópa, stærri en röt og minni en röt. Til að finna hlut í trénu er byrjað í röt og svo minnkad leitarsvæðið að hlutfénu sem á við. Ef leitarstakið er minna en röt er farið til vinstri annars til hægri.

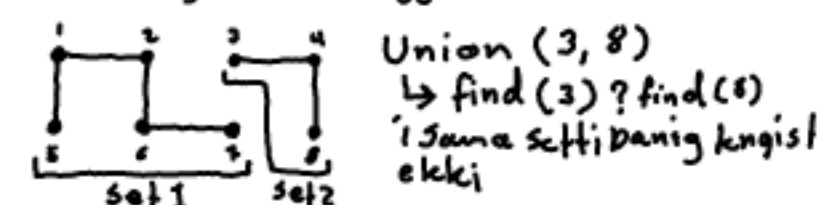
$A[2, 5, 6, 8, 9, 11] M = A[\text{length}/2]$

Byrjað á að finna miðju (M) og hugsað um sem röt trésins. Afmarkar svo leitarsvæðið við topp og botn.

Union find og elu:

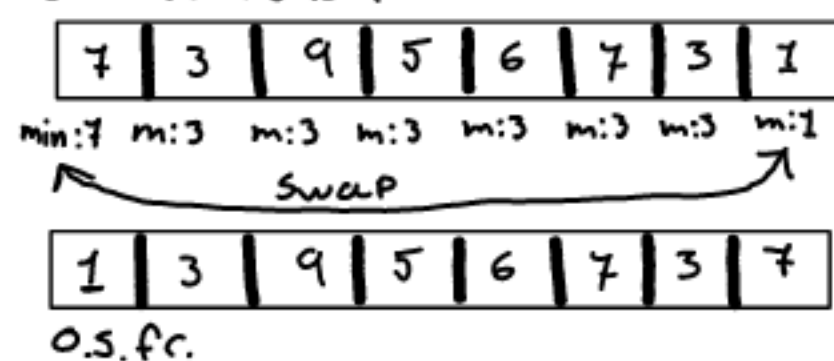
Notað á hóp af ótengdum hlutum.

- Athugasemur hvort tveir hlutir tilheyri sama hóp
- Tengir (union) tvo hluti ef þeir tilheyra sama hóp
- Gefur séð hvort hlutir myndu hring ef þeir myndu tengjast

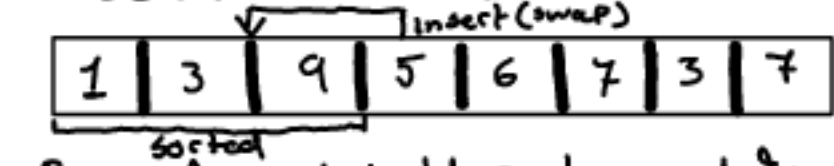


Selection sort:

Fær í gegnum heilt fylki hlutur utan um minnstastakið fyrir hvert skref og swappar við í



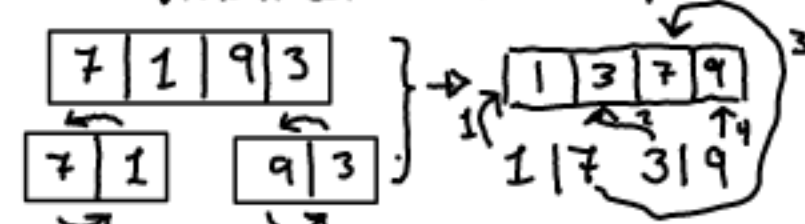
Insertion sort:



Byrjar fremst heldur utan um það sem búið er að sortu. Ef hlutur er minni en stak á undan, líkir fram á þangað til hlutur er stærri en stak á undan. Setur þá inn og heldur áfram frá stakinu sem var swappað.

Merge sort:

Bútar fylki niður í helminga þangað til meður er bara með tvö stök. Þau eru borið saman og sett saman í bútafylki. Bútafylkið er svo borið saman við nærliggjandi bútafylki. Afsinnustærð og Mergeð saman. Þessi virkni heldur áfram upp tréð.

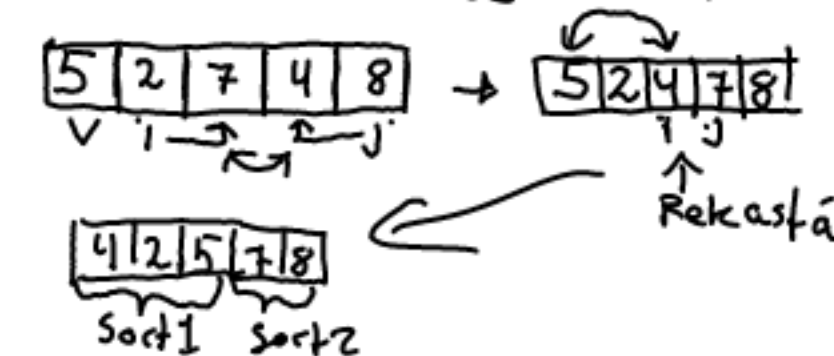


Merge ferur í sér að velja lægsta stakið úr báðum fylkingum, bera þau saman og setja minna stakið inn í Merge fylkið.

Quicksort:

Velur stak af handahófi og raðar öðrum stökum í kringum það, minni og stærri, þessi partar af fylkinu eru svo sortaðir innbyrðis.

Stærsti partur aðferðarinnar er að skipta fylkinu. Þegar VMS er valdið er það sett fremsi og tveir pointerar settið í gang. Í sum leitar frá vinstri að stökum stærri en VMS, hinna, leita frá hægri að stökum minni en VMS. Þegar báðir hafa fundið stak er þeim víxlað og útað að næsta færi $i+1$ og $j-1$ til að finna ekki sama. Þegar engin stök til að víxla er veltur, í og úr rekast á

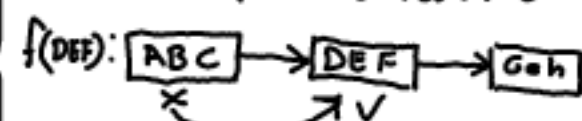


Heapsort:

Setur öll stök inn í Min-PQ þannig minsta stakið sé efst og raðar svo aftur inn í upprunalegt fylki með því að farlægja minnstastakið úr PQ

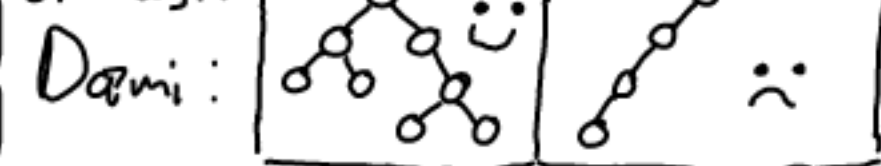
Sequential search:

Defaultið fyrir linked lista. Leitar að staki með því að fara í gegnum öll stök listans og athugasemur hvort gildin stemmi við leit ef ekki fara án ærsta.



BST (Binary Search Tree):

Basicið, Binary tré hægti stærra, vinstri minna. Þar fæst að vera balanced svo getur lent í að öll stök séu vinstri eða öll hægri



Red Black-BST:

Hér er tréð "self balancing". Hæðin má ekki og "getur" ekki orðið meiri en $2 \lg(n)$ fyrir n hnúta.

Hugsanum þetta á 2-3 forni, allt af setja eins niðarlega og hægt er og ef hnútur heldur 3 að þá er taí lauf og losa upp

Hashing:

Hash nota tvo hluti til að raða í töflu, Hash function t.d. $h(k) = 3k + k^2$ og lengd töflu. Útkoman úr $h(k)$ fyrir lykil k er moduloð, %, við lengd töflunnar og k setti $A[h(k) \% m]$

Áreksstrum er reðlað á tvö mismunandi vegu. Separate chaining, þar sem hvert stak er linked listi. Árekskur bætir stakið $next$ á núverandi key. Linear Probing ef árekskur þá leitast að næsta lausa plássi og merkja árekskurs höfðið sem notað svo ef það er tekið útleikkar leitun ekki

Depth First Search (DFS):

Notar stack og recursion til að finna leið frá punkti X til Y. Hraður og greedy.

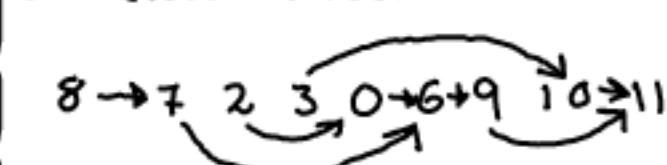
Merkir hnútum leið og fer fram hjá, bætir aðliggjandi hnútum á stack, poppar stakið og kallar á DFS með pop stakinu... recurse

Breadth First Search (BFS):

Mjög svipað DFS, notar ekki recursion, notar queue í stað stack og finnur stöfu leiðir í övrigtudu neti.

Topological Order:

Er aðeins hægt að útfæra í stefnum acyclic netum. Hnútum er raðað í röð þannig þeir bændi allir til hægri á hnút sem eru á eftir sér í röðinni



Prim's Algorithm:

Byrjar á ákupunkt og merkir þyngd og leið allra aðliggjandi leggja í Min-PQ. Fer svo á þann hnút sem liggur handan leggins með minnstu þyngd og bætir aðliggjandi leggjum þess í PQ-ið. Þetta er svo endurtekið V-1 sinnum. Virkar betur á net með mihið af leggjum.

Kruskal's Algorithm:

Byrjar á að setja alla legg í Min-PQ. Velur minnstu legginu og handir úr PQ. Athugasemur hvort minna minnsti leggur myndi hring ef tengdur, Union find, ef enginn hringur tengja bitchið Bada bing Bada boom

Regex

*	0 eða fleiri
+	1 eða fleiri
?	0 eða 1
{3}	nákvæmlega 3
{3,}	3 eða fleiri
{3,5}	3, 4 eða 5

.	hvaða stakur sem er
(a b)	a eða b
(...)	Höpur, tekið saman

amk tvö 0, ekki röð
 $1*01+0(1+0?)*$
amk þrjú 0, ekki röð
 $1*(01+){2,}0(1+0?)*$



tímar - sorting

reiknirit	P	S	best	avg	worst	viðbót
selection	x	-	n^2	n^2	n^2	lélegt
insertion	x	x	n	n^2	n^2	nota fyrir lítil hálf sortuð
merge	-	x	$n \log(n)$	$n \log(n)$	$n \log(n)$	alltaf $n \log(n)$
quick	x	-	$n \log(n)$	$n \ln(n)$	n^2	nánast alltaf $n \log(n)$
heap	x	-	n	$n \log(n)$	$n \log(n)$	alltaf allavega $n \log(n)$

tímar - gagnasöfnin

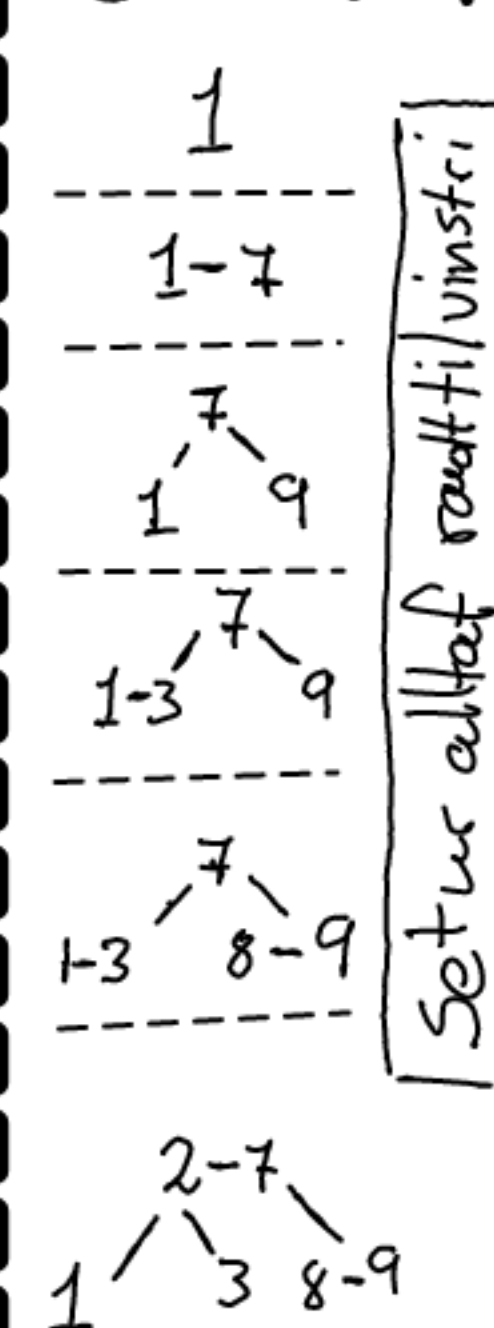
gagnasafn	search (worst)	in (worst)	del (worst)	search (avg)	in (avg)	del (avg)
sequential	n	n	n	n	n	n
binary search	$\log(n)$	n	n	$\log(n)$	$\log(n)$	$\log(n)$
BST	n	n	n	$\log(n)$	$\log(n)$	\sqrt{n}
red-black-BST	$\log(n)$	$\log(n)$	$\log(n)$	$\log(n)$	$\log(n)$	$\log(n)$
HT - SH	n	n	n	1	1	1
HT - LP	n	n	n	1	1	1

tímar allskonar netashit

vandamál	reiknirit	tími	pláss
path	DFS	$E + V$	V
shortest path (fewest edges)	BFS	$E + V$	V
cycle	DFS	$E + V$	V
directed path	DFS	$E + V$	V
shortest directed path (fewest edges)	BFS	$E + V$	V
directed cycle	DFS	$E + V$	V
topological sort	DFS	$E + V$	V
bipartiteness / odd cycle	DFS	$E + V$	V
connected components	DFS	$E + V$	V
Eulerian cycle	DFS	$E + V$	$E + V$
directed Eulerian cycle	DFS	$E + V$	V
transitive closure	DFS	$V(E + V)$	V^2
minimum spanning tree	Kruskal	$E \log E$	$E + V$
minimum spanning tree	Prim	$E \log V$	V
shortest paths (nonnegative weights)	Dijkstra	$E \log V$	V
shortest paths (no negative cycles)	Bellman-Ford	$V(V + E)$	V
shortest paths (no cycles)	topological sort	$V + E$	V

$A = [1, 7, 9, 3, 8, 2]$

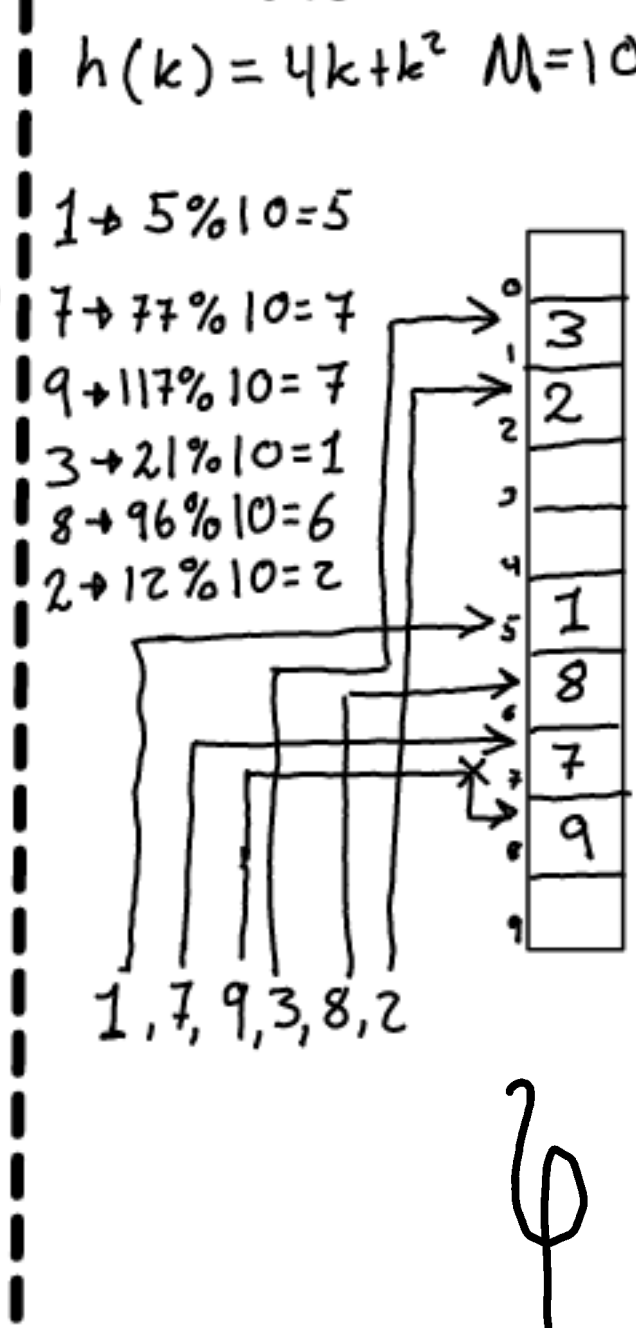
LL-RB-BST



Kruskal's



LP-Hash table



tumi