

Lecture 12: Sorting Wrap Up

Bucket Sort: If only k "types" appear in your list, make k "buckets"

$[1, 0, 1, 0, 2, 2, 1, 3, 1, 0]$

↓

Step 1: place into
buckets

$O(n)$



Step 2:

$O(n+k)$

↓

$[0, 0, 0, 1, 1, 1, 1, 2, 2, 3]$

In place? **NO**

Adaptive? **NO**

Stable? **YES**

Radix Sort

radix =

idea:

example: (base/radix = 10)

[103, 801, 800, 401, 101, 901, 018]

1's digit

--	--	--	--	--	--	--	--	--	--	--

10's digit

--	--	--	--	--	--	--	--	--	--	--

100's digit

--	--	--	--	--	--	--	--	--	--	--

Key property of Bucket Sort needed?

runtime?

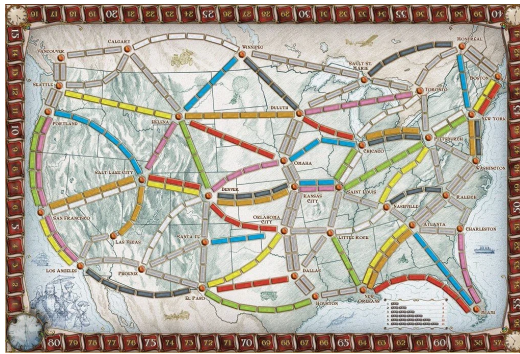
example: suppose we know each element of our size n list is a positive integer smaller than $n^{100\sqrt{n}}$.

Bucket?

comparison-based?

radix?

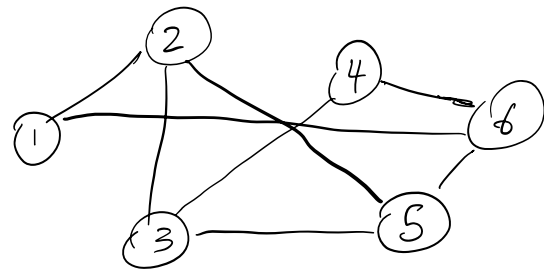
GRAPHS



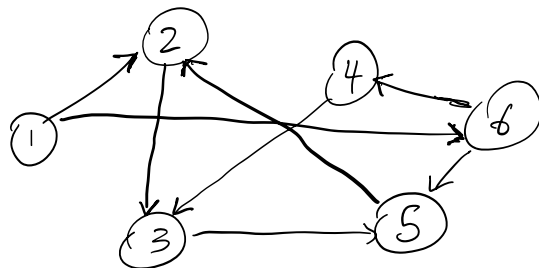
Def (Undirected)

degree

neighbors/adjacent



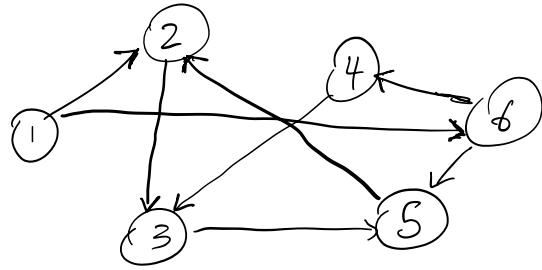
Directed:



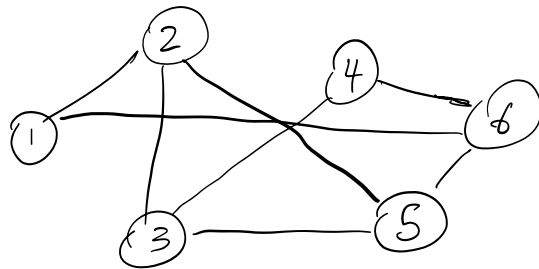
"in-degree"

"out-degree"

non-simple graphs



weighted graphs



Applications?

what are nodes/edges? directed? simple? weighted?

- facebook friends
- Java Inheritance
- Airline/train routes
-

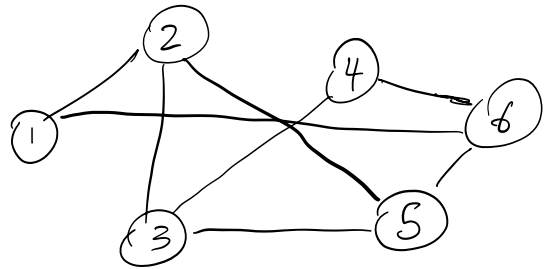
Complete Graphs

undirected

directed

directed, non-simple

Def (path)



Simple path

cycle

"six degrees of separation"

Def (strongly connected)

weakly connected

maximum # of edges?

undirected, simple:

directed, simple:

dense:

sparse:

question: if G connected, min # of edges?

In general, $|E|$ and $|V|$ very different, so runtime should be a function of
both $|E|$ and $|V|$

Trees