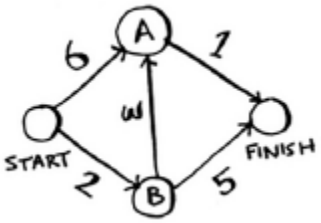
Chapter 7

Dijkstra's Algorithm

In the previous chapter we talked about how to find the shortest path in a graph using Breadth-First Search, but that only works on unweighted graph which means there is no cost included and we are only looking for the smallest number of edges, for the weighted graphs we need Dijkstra's Algorithm.

Steps:



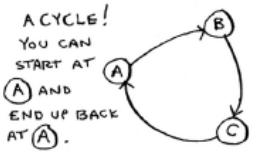
NODE	TIME TO NODE
A	6
B	2
FINISH	∞

1. Find the cheapest node to go to from the starting point, it takes 6 minutes to get to node A and 2 minutes to get to node B. The rest of the nodes, you don't know yet.
2. Update the costs of the neighbors of this node, calculate how long it takes to get to all of node B's neighbors by following an edge from B.
3. Repeat until you've done this for every node in the graph.
4. Calculate the final path.

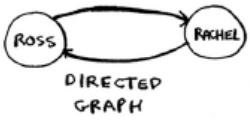
NODE	TIME
A	~~6~~ 5
B	2
FINISH	7

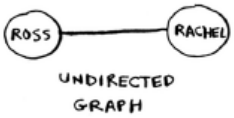
Terminologies:

- In weighted graphs each edge has a number it is associated with called weight or cost as in the fig. above.
- These weights affect our choice for the shortest path.
- A graph where you can end at the starting point is called cycle.



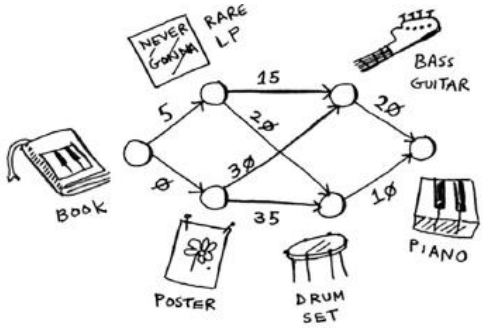
- In directed graphs edges have specific direction from a specific node to another, but in undirected edges just connect 2 nodes.





- A cycle graph between two nodes and an undirected edge between them (as in the previous figure) have the same meaning.

Example:

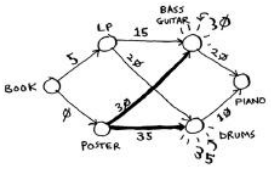


in this example we need to find the cheapest way to get a piano by trading starting with a book, first get the costs of the starting point neighbor and find the smallest, after that update the values of that node neighbors

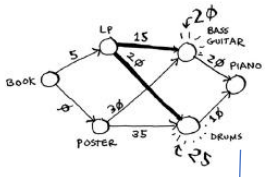
Now repeat the last two steps until all the nodes are done.

NODE	COST
LP	5
POSTER	∅
GUITAR	∞
DRUMS	∞
PIANO	∞

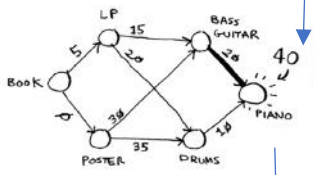
WE HAVEN'T REACHED THESE NODES FROM THE START YET



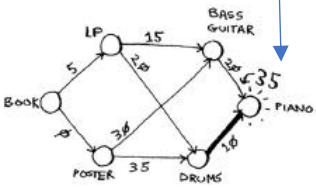
PARENT	NODE	COST
BOOK	LP	5
BOOK	POSTER	∅
~~POSTER~~	GUITAR	~~∅∅~~ ~~30~~
~~POSTER~~	DRUMS	~~∅∅~~ ~~35~~
—	PIANO	∞



PARENT	NODE	COST
BOOK	LP	5
BOOK	POSTER	∅
~~LP~~	GUITAR	~~∅∅~~ ~~20~~
~~LP~~	DRUMS	~~∅∅~~ ~~25~~
—	PIANO	∞

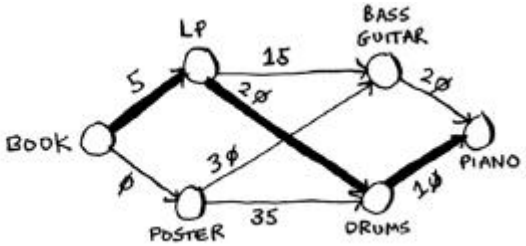


PARENT	NODE	COST
BOOK	LP	5
BOOK	POSTER	∅
LP	GUITAR	20
LP	DRUMS	25
GUITAR	PIANO	~~40~~



PARENT	NODE	COST
BOOK	LP	5
BOOK	POSTER	∅
LP	GUITAR	20
LP	DRUMS	25
DRUMS	PIANO	~~35~~

Finally we can get to see how the final path looks like



Note: negative weights doesn't work with Dijkstra, it makes you take a longer path.