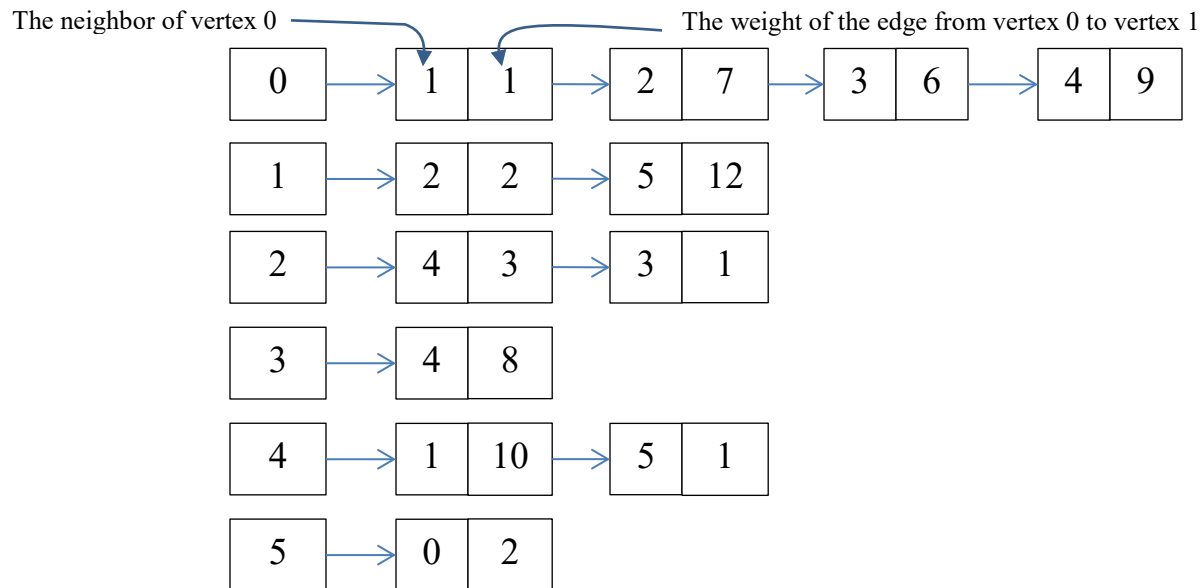


TIC2001 Data Structure and Algorithm Quiz 3

Question 1 (10 marks)

Here is a graph (with 6 vertices) represented by its adjacency list. In the adjacency list, each node will point to another node with the left entry is the vertex index and the right value is the weight of the edge.



Perform Dijkstra Algorithm of this graph to find all the shortest distance from the node 0. Each of the following table is a priority queue sorted by the shortest estimated distance, $\delta(0,v)$. **You have to sort the priority queue by putting the node with the least estimated distance on top.**

Node v	$\delta(0,v)$		Node v	$\delta(0,v)$		Node v	$\delta(0,v)$		Node v	$\delta(0,v)$	
0	0										
1	∞										
2	∞	Extract			Extract			Extract			Extract
3	∞	0			—			—			—
4	∞	→			→			→			→
5	∞										

Node v	$\delta(0,v)$		Node v	$\delta(0,v)$		Node v	$\delta(0,v)$		Node v	$\delta(0,v)$	
		Extract			Extract			Extract			Extract
		—			—			—			—
		→			→			→			→

Write down the final shortest distances for each vertex from node 0:

Vertex 0: _____ Vertex 1: _____ Vertex 2: _____ Vertex 3: _____ Vertex 4: _____ Vertex 5: _____

Question 2 Currency Conversion (10 marks)

Earlier in this year, my family went to Taiwan and we needed some Taiwan dollars (TWD). However, we found out that it is better to convert our Singapore dollars (SGD) to American dollars (USD) first, and then convert our USD into TWD. In this way, we resulted in more Taiwan dollars at the end, when comparing to converting SGD into TWD directly.

If we have the conversion matrix like the one below and we would like to convert a certain currency A to B, how do we find out what is the best way to maximize our final amount of currency B?

	USD	Euro	GBP	SGD
USD	1	0.75	0.63	1.26
Euro	1.33	1	0.84	1.67
GBP	1.55	1.18	1	2.00
SGD	0.79	0.59	0.49	1

So, if we have altogether k types of currencies, $r[i,j]$ will be the conversion rate from currency i to currency j . Assuming that there is no *arbitrage**, i.e., converting from A to B to C to A will always get you no more money than you started with. (It may get you less, however.) Also, there may be some conversions that are impossible, e.g. you cannot exchange A for B.

Explain briefly how you will model the problem.

Explain in less than or equal to three sentences which algorithm you will use, and how to use it.

What is the running time of your algorithm in terms of k ?

What if we allow arbitrage to happen? How does it destroy your suggested algorithm and how should we detect it?

* *Arbitrage: the simultaneous buying and selling of securities, currency, or commodities in different markets or in derivative forms in order to take advantage of differing prices for the same asset.*