# Computer Systems Tutorial

December $8^{th}$ 2023

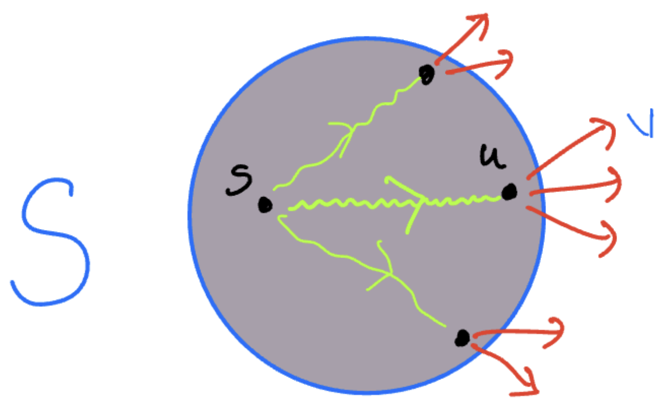
## 1 Preliminaries

**Graph** A graph is a pair of two sets: a set of *edges* and a set of *verticies* or nodes. We will denote it like $(V, E)$ where elements of $E$ are also pairs like $(u, v)$ that stands for an edge from vertex $u$ (the source) to vertex $v$ (the target). We will then consider weighted graphs such that edges are of the form $(u, w, v)$ where $w$ denotes the weight of an edge, e.g. how 'costly' it is to follow that particular edge. A *path* from vertex $u$ to vertex $v$ in a graph is a sequence of nodes $(v_1, \ldots, v_n)$ such that the first vertex in the sequence $v_1 = u$ and the last vertex is $v_n = v$ and there is an edge between $v_i$ and $v_{i+1}$ for all $i < n$.

**Dijkstra Algorithm** This algorithm finds the distances of shortest paths from one source vertex $v$ to all other vertices in a weighted graph such that all weights are positive. The pseudo-code is given below.

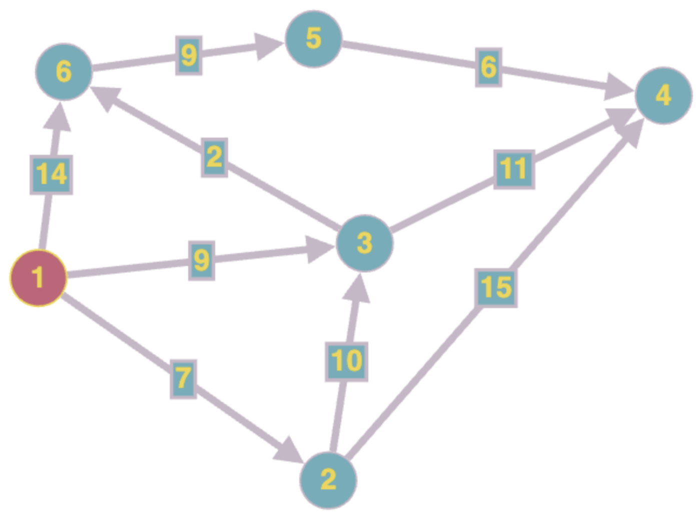
```
# s is the source vertex
    fun dijkstra(s):
    # D stores distances from s to all other verticies
        D = {}
        S = {s}

        for v in V:
            if (s,v) in E:
                D[v] = c(s,v) # c is the cost of the edge
            else:
                D[v] = float('inf')

        while S != V:
            v = min(D) and v not in S
            S.add(v)

            for (v,u) in E:
                D[u] = min(D[v] + c(v,u), D[u])
```

$S$ is the set of currently visited vertices $v'$ and it is guaranteed that the distance from $s$ to $v' \in S$ is the shortest. Then the algorithm picks a vertex $v$ not from $S$ such that the distance to it from the source node is minimal across all candidates and updates the distances to all $v$'s neighbours.




## 2 Exercises

**Exercise 1** Find the distances of all shortest paths from source 1 in the graph below by using Dijkstra algorithm.




What is the set $S$ at each step? What are the distances at each step?

**Exercise 2** What is the complexity of the above algorithm in terms of $V$ and $E$?

**Exercise 3** Consider the following set of processes:

| Process | Arrival Time | Burst time |
|---|---|---|
| $P_1$ | 0 ms | 5 ms |
| $P_2$ | 2 ms | 8 ms |
| $P_3$ | 3 ms | 6 ms |
| $P_4$ | 5 ms | 2 ms |

What is the average response time assuming Round Robin CPU scheduling policy with a Time Quantum (q = 4 ms). Also, note that newly arriving processes are added to the tail of the ready queue. The response time is the time it takes for the process to be first allocated a CPU after it arrives.

**Exercise 4** Suppose Host A wants to send a large file to Host B. The path from Host A to Host B has four links, of rates R1 = 12 Mbps, R2 = 15 Mbps, R3 = 8 Mbps and R4 = 25 Mbps. What will be the minimum end-to-end delay of transferring a video clip of 535 MegaBytes? (We assume that there is no store-and-forward delay on the routers connecting the links)

**Exercise 5** In the **SYNACK** message of the TCP connection establishing handshake, the server sends a 1-byte TCP segment to the client. That segment has Sequence number = 900, SYN/ACKbits = 1. Which of the following TCP headers are correct for the corresponding **ACK** message from the client to the server?

a) Sequence number = 901, Acknowledgement number = 901, SYNbit = 0, ACKbit = 1

b) Sequence number = 500, Acknowledgement number = 901, SYNbit = 0, ACKbit = 1

c) Sequence number = 901, Acknowledgement number = 483, SYNbit = 0, ACKbit = 1

d) Sequence number = 483, Acknowledgement number = 901, SYNbit = 1, ACKbit = 1

e) Sequence number = 901, Acknowledgement number = 901, SYNbit = 1, ACKbit = 1

f) Sequence number = 901, Acknowledgement number = 483, SYNbit = 1, ACKbit = 1